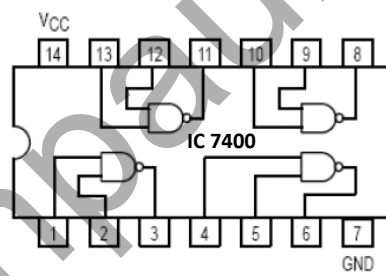
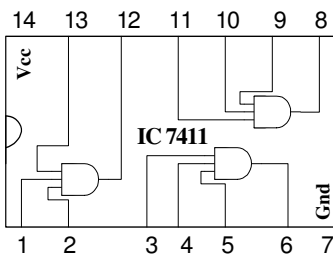
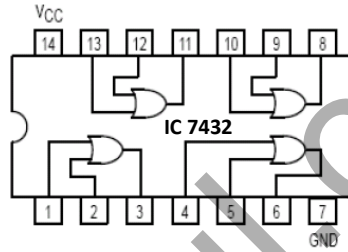
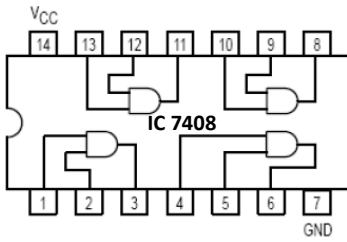
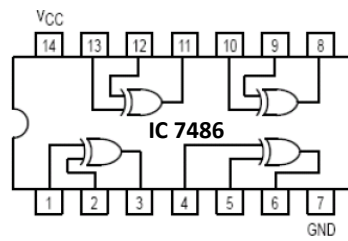
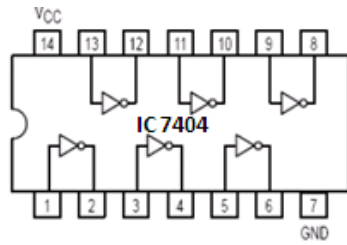


Pin Diagram of Basic Logic Gates:



www.rejinpaul.com

EX: NO:

DATE:

Design and Implementation of Adder:

Aim:

To design, Implement and verify the truth table of adder using logic gates.

Apparatus Required:

Sl. No.	COMPONENT	SPECIFICATION	QTY
1.	AND GATE	IC 7408	1
2.	OR GATE	IC 7432	1
3.	X-OR GATE	IC 7486	1
4.	DIGITAL IC TRAINER KIT	-	1
5.	PATCH CORD	-	-

Theory:

Half Adder:

A half adder is a combinational circuit that performs the sum of two binary digits (A, B) to give a maximum of two binary outputs namely the sum(S) and the carry(C). Carry is the higher order bit and the sum is the lower order bit of the output. The Boolean expression for the sum(S) and carry(C) of half adder is,

$$S = A \oplus B \quad C = AB$$

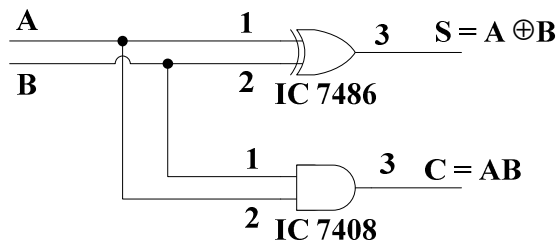
Full Adder:

A full adder is a combinational circuit that performs the sum of three binary digits (A, B, Cin) to give a maximum of two binary outputs namely the sum(S) and the carry-out (Cout). The full adder becomes necessary when a carry input must be added to the two binary digits to obtain the correct sum. A half adder has no input for carries from previous circuits. The Boolean expression for the sum (S) and carry-out (Cout) of full adder is,

$$S = A \oplus B \oplus C_{in} \quad C_{out} = AB + AC_{in} + BC_{in}$$

Logic Diagram:

Half Adder:



Truth Table:

A	B	CARRY(C)	SUM(S)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

K-Map for Sum:

	B	0	1
A	0		1
	1	1	

$SUM(S) = A \oplus B$

K-Map for Carry:

	B	0	1
A	0		
	1		1

$CARRY(C) = AB$

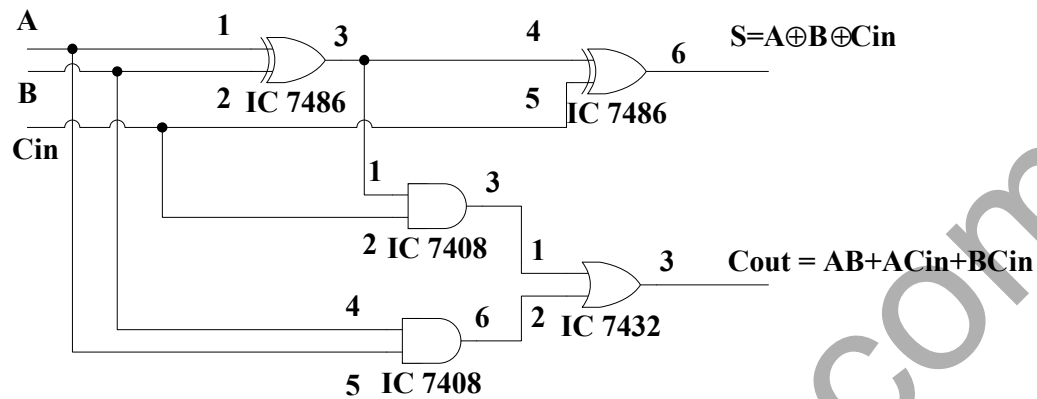
Procedure:

1. Verify the gates.
2. Make the connections as per the circuit diagram.
3. Switch on VCC and apply various combinations of inputs according to truth table.
4. Note down the output readings of sum and the carry bit for different combinations of inputs.

www.rejinpaul.com

Logic Diagram:

Full Adder:



Truth

Table:

A	B	Cin	CARRY(Cout)	SUM(S)
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

K-Map for Sum:

A \ BCin	00	01	11	10
	0	1		1
1	1		1	

K-Map for Carry:

A \ BCin	00	01	11	10
	0		1	
1		1	1	1

$$\begin{aligned}\text{Sum (S)} &= A \oplus B \oplus C_{in} \\ \text{CARRY (Cout)} &= AB + BC_{in} + AC_{in}\end{aligned}$$

$$AB + BC_{in} + AC_{in}$$

Result:

Thus the adder was designed and implemented with their truth table verified using logic gates.

www.rejinpaul.com

www.rejinpaul.com

EX: NO:

DATE:

Design and Implementation of Subtractor:

Aim:

To design, Implement and verify the truth table subtractor using logic gates.

Apparatus Required:

Sl. No.	COMPONENT	SPECIFICATION	QTY
1.	NOT GATE	IC 7404	1
2.	AND GATE	IC 7408	1
3.	OR GATE	IC 7432	1
4.	X-OR GATE	IC 7486	1
5.	DIGITAL IC TRAINER KIT	-	1
6.	PATCH CORD	-	-

Theory:

Half Subtractor:

A half subtractor is a combinational circuit that performs the difference between two binary digits (A, B) to give a maximum of two binary outputs namely the Difference (Diff) and the Borrow (Brw). The Borrow output here specifies whether a '1' has been borrowed to perform the subtraction. The Boolean expression for the difference (Diff) and borrow (Brw) of half subtractor is,

$$\text{Diff} = A \oplus B \quad \text{Brw} = A'B$$

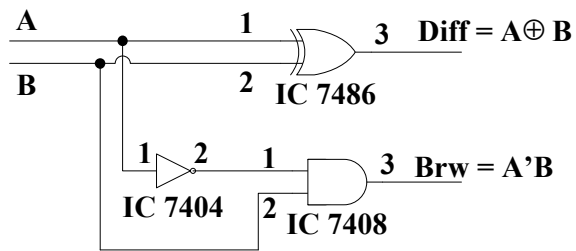
Full Subtractor:

A full subtractor is a combinational circuit that performs the three bit subtraction (A, B, Bin) to give a maximum of two binary outputs namely the difference (Diff) and the borrow (Bout). Full subtractor takes into consideration whether a '1' has already been borrowed by the previous adjacent lower minuend or not. The Boolean expression for the difference (Diff) and borrow (Bout) of full subtractor is,

$$\text{Diff} = A \oplus B \oplus \text{Bin} \quad \text{Bout} = A'B + A'\text{Bin} + B\text{Bin}$$

Logic Diagram:

Half Subtractor:



Truth Table:

A	B	BORROW (Brw)	DIFFERENCE (Diff)
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

K-Map for Difference:

A \ B	0	1
0		1
1	1	

Difference (Diff) = $A \oplus B$

K-Map for Borrow:

A \ B	0	1
0		1
1		

Borrow (Brw) = $A'B$

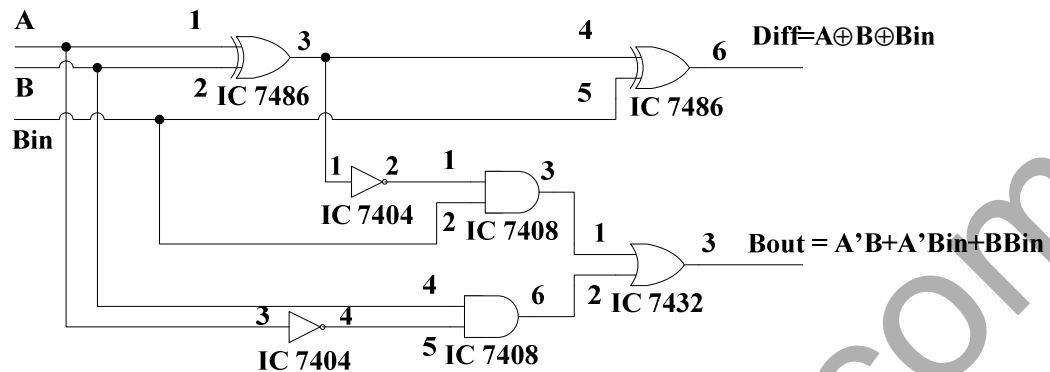
Procedure:

1. Verify the gates.
2. Make the connections as per the circuit diagram.
3. Switch on VCC and apply various combinations of input according to truth table.
4. Note down the output readings for difference and the borrow bit for different combination of inputs.

www.rejinpaul.com

Logic Diagram:

Full Subtractor:



Truth Table:

A	B	Bin	BORROW (Bout)	DIFFERENCE (Diff)
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

K-Map for Difference:

Bin	A=0	A=1
0	0	1
1	1	1

$\oplus \text{ Bin}$

K-Map for Borrow:

Bin	A=0	A=1
0	0	1
1	0	1

Borrow (Bout) = $A'B + A'\text{Bin} + B\text{Bin}$

Difference (Diff)
= $A \oplus B$

Result:

Thus the subtractor was designed and implemented with their truth table verified using logic gates.

www.rejinpaul.com

www.rejinpaul.com

EX: NO:

DATE:

Design and Implementation of BCD to Excess-3 and Excess-3 to BCD

Code Converters:

Aim:

To design, implement and verify the following code converters using logic gates,

- a) BCD to Excess-3 code converter
- b) Excess-3 to BCD code converter

Apparatus Required:

SI. No.	COMPONENT	SPECIFICATION	QTY
1.	NOT GATE	IC 7404	1
2.	AND GATE	IC 7408	2
3.	OR GATE	IC 7432	1
4.	X-OR GATE	IC 7486	1
5.	3 INPUT AND GATE	IC 7411	1
6.	DIGITAL IC TRAINER KIT	-	1
7.	PATCH CORD	-	-

Theory:

Code Converters:

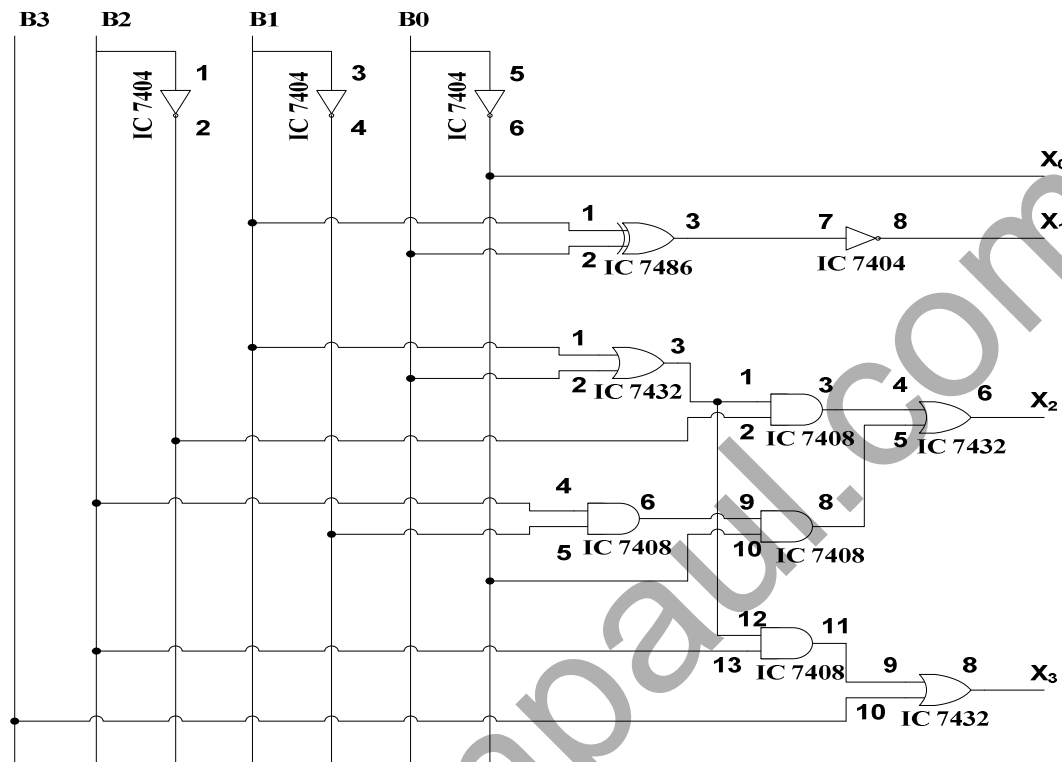
The availability of large variety of codes for the same discrete elements of information results in the use of different codes by different systems. A conversion circuit must be inserted between the two systems if each uses different codes for same information. Thus, code converter is a circuit that makes the two systems compatible even though each uses different binary code.

Binary Coded Decimal:

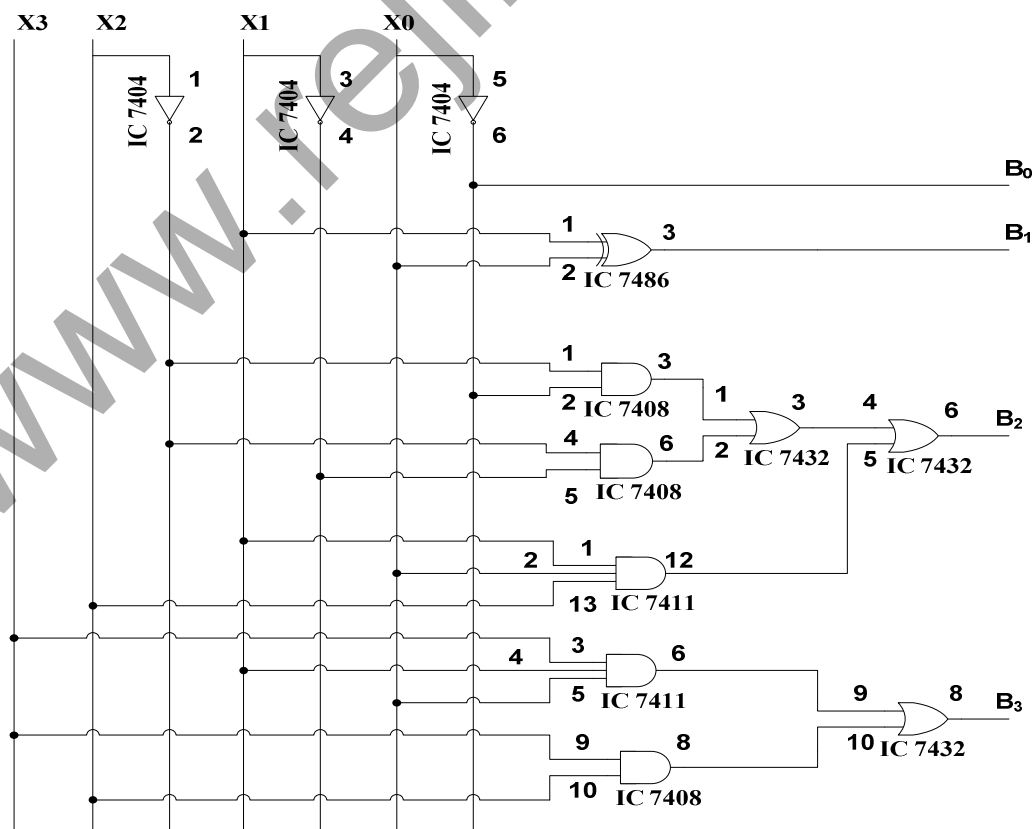
Binary Coded Decimal is a method of using binary digits to represent the decimal digits 0 through 9. It is possible to assign weights to the binary bits according to their positions. The weights in the BCD code are 8, 4, 2 and 1. Ex: $(127)_{10}$ - BCD equivalent $(0001\ 0010\ 0111)_2$.

Logic Diagram:

BCD to Excess-3 Code Converter:



Excess-3 to BCD Code Converter:



Excess-3 Code:

This is an un-weighted code. Its code assignment is obtained from the corresponding value of BCD after the addition of $(0011)_2$.

BCD to Excess-3 (or) Excess-3 to BCD:

Since each code uses four bits to represent a decimal digit, there must be four inputs and four output variables. Four binary variables have sixteen different input combinations, only ten of the input combinations are listed in the truth table. The six bit combinations not listed for the input variables are don't care combination. For BCD to Excess-3, the input variables are designated as B_3, B_2, B_1, B_0 and the output variables are designated as X_3, X_2, X_1, X_0 in the truth table. The Boolean functions are obtained from K-Map for each output variable. The combinational logic for the code converters are designed according to the Boolean expressions from K-Map simplification. The Boolean expressions from the K-Map are shown below.

Boolean Expression for BCD to Excess-3 Code Conversion:

$$X_3 = B_0B_2 + B_1B_2 + B_3$$

$$X_2 = \bar{B}_0\bar{B}_2 + B_1\bar{B}_2 + \bar{B}_0B_1B_2$$

$$X_1 = \overline{B_0 \oplus B_1}$$

$$X_0 = \bar{B}_0$$

Boolean Expression for Excess-3 to BCD Code Conversion:

$$B_3 = X_0X_1X_3 + X_2X_3$$

$$B_2 = X_0\bar{X}_2 + X_0X_1X_2 + \bar{X}_1\bar{X}_2$$

$$B_1 = X_0 \oplus X_1$$

$$B_0 = \bar{X}_0$$

Procedure:

1. Verify the gates.
2. Make the connections as per the circuit diagram.
3. Switch on VCC and apply various combinations of input according to truth table.
4. For all input combinations the outputs are verified with the truth table.

Truth Table for BCD to Excess-3 code and Vice Versa:

BCD input				Excess – 3 output			
B3	B2	B1	B0	X3	X2	X1	X0
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

K-Map for BCD to Excess-3 Code Converter:

K-Map for X0:

B3 B2	B1 B0			
	00	01	11	10
00	1			1
01	1			1
11	X	X	X	X
10	1		X	X

$$X0 = B0'$$

K-Map for X1:

B3 B2	B1 B0			
	00	01	11	10
00	1		1	
01	1		1	
11	X	X	X	X
10	X		X	X

$$X1 = B1 \oplus \overline{B0}$$

www.rejinpaul.com

K-Map for X2:

B1 B0		B3 B2			
		00	01	11	10
B3 B2	00		1	1	1
	01	1			
	11	X	X	X	X
	10		1	X	X

$$X2 = B2'(B0+B1) + B2B1'B0'$$

K-Map for X3:

B1 B0		B3 B2			
		00	01	11	10
B3 B2	00				
	01		1	1	1
	11	X	X	X	X
	10	1	1	X	X

$$X3 = B3 + B2(B0+B1)$$

K-Map for Excess-3 to BCD Code Converter:

K-Map for B0:

X1 X0		X3 X2			
		00	01	11	10
X3 X2	00	X	X		X
	01	1			1
	11	1	X	X	X
	10	1			1

$$B0 = X0'$$

K-Map for B1:

X1 X0		X3 X2			
		00	01	11	10
X3 X2	00	X	X		X
	01		1		1
	11		X	X	X
	10		1		1

$$B1 = X1 \oplus X0$$

K-Map for B2:

X1 X0		X3 X2			
		00	01	11	10
X3 X2	00	X	X		X
	01			1	
	11		X	X	X
	10	1	1		1

$$B2 = X2'X0' + X2'X1' + X2X1X0$$

K-Map for B3:

X1 X0		X3 X2			
		00	01	11	10
X3 X2	00	X	X		X
	01				
	11	1	X	X	X
	10			1	

$$B3 = X3X1X0 + X3X2$$

Result:

Thus BCD to Excess-3 and Excess-3 to BCD converters are designed, constructed using logic gates and their truth table was verified.

www.rejinpaul.com

www.rejinpaul.com

EX: NO:

DATE:

Design and Implementation of Binary to Gray and Gray to Binary

Code Converters:

Aim:

To design, implement and verify the following code converters using logic gates,

- a) Binary to Gray code converter
- b) Gray to Binary code converter

Apparatus Required:

SI. No.	COMPONENT	SPECIFICATION	QTY
1.	X-OR GATE	IC 7486	1
2.	DIGITAL IC TRAINER KIT	-	1
3.	PATCH CORD	-	-

Theory:

Code Converters:

The availability of large variety of codes for the same discrete elements of information results in the use of different codes by different systems. A conversion circuit must be inserted between the two systems if each uses different codes for same information. Thus, code converter is a circuit that makes the two systems compatible even though each uses different binary code.

Gray Code:

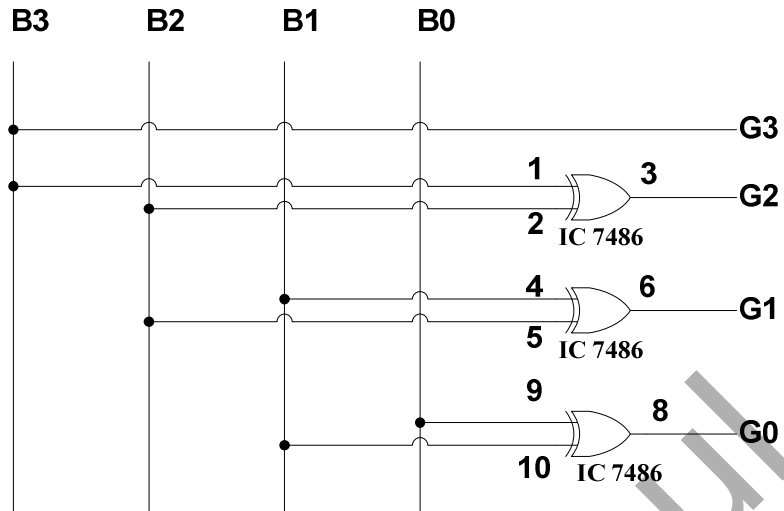
To obtain a different gray code, one can start with any bit information and proceed to obtain the next bit combination by changing only one bit from 0 to 1 (or) 1 to 0 in any desired random fashion provided any two numbers do not have identical code assignments.

Binary to Gray (or) Gray to Binary:

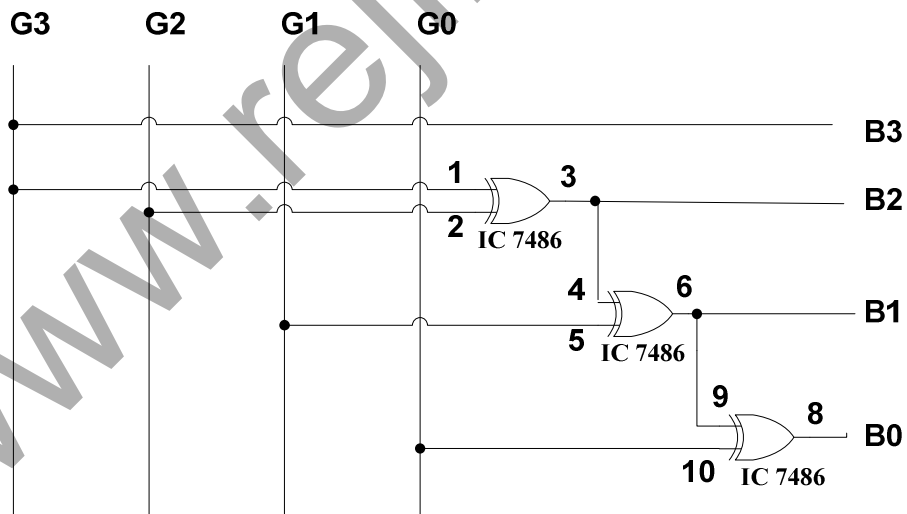
To convert from binary code to Gray code, the input lines must supply the bit combination of elements as specified by the code and the output lines generate the corresponding bit combination of code. Each one of the four maps represents one of the four outputs of the circuit as a function of the four input variables.

Logic Diagram:

Binary to Gray Code Converter:

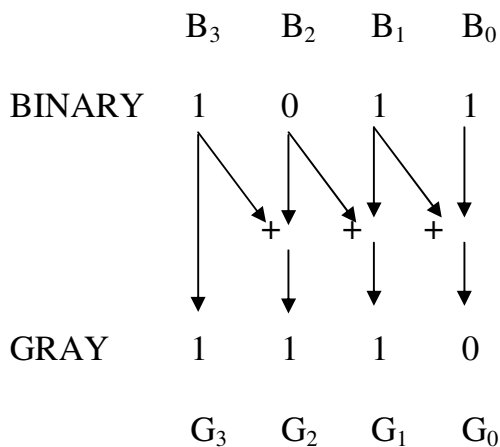


Gray to Binary Code Converter:



Binary to Gray Code Conversion Steps:

The example shows the steps involved in conversion of a binary code to its gray code. Binary code taken for the example is 1011.



In the conversion process the most significant bit (MSB) of the binary code is taken as the MSB of the Gray code. The bit positions G₂, G₁ and G₀ is obtained by adding (B₃, B₂), (B₂, B₁) and (B₁, B₀) respectively, ignoring the carry generated. From the K-Map simplification for binary to Gray code conversion the following Boolean expressions are obtained,

$$G_3 = B_3$$

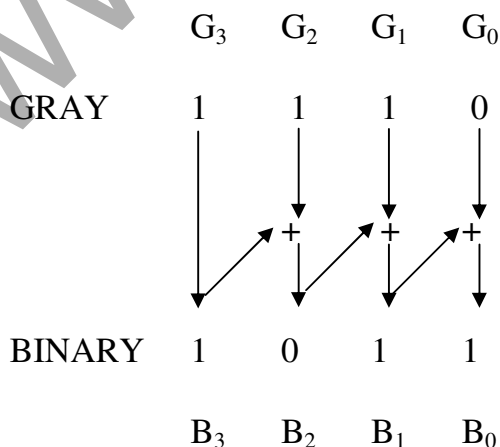
$$G_2 = B_3 \oplus B_2$$

$$G_1 = B_2 \oplus B_1$$

$$G_0 = B_1 \oplus B_0$$

Gray to Binary Code Conversion Steps:

The example shows the steps involved in conversion of a Gray code to binary code. Gray code taken for the example is 1110.



Truth Table for Binary to Gray code and vice versa:

Binary input				Gray output			
B3	B2	B1	B0	X3	X2	X1	X0
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 for Binary to Gray Code Converter:

K-Map for G0:

		B1 B0			
B3 B2		00	01	11	10
		00	1		1
	01		1		1
	11		1		1
	10		1		1

$$G0 = B1 \oplus B0$$

K-Map for G1:

		B1 B0			
B3 B2		00	01	11	10
				1	1
	01	1	1		
	11	1	1		
	10			1	1

$$G1 = B1 \oplus B2$$

In the conversion process the most significant bit (MSB) of the Gray code is taken as the MSB of the binary code. The bit positions B2, B1 and B0 is obtained by adding (B3, G2), (B2, G1) and

(B1, G0) respectively, ignoring the carry generated. From the K-Map simplification for Gray code to binary code conversion the following Boolean expressions are obtained,

$$B_3 = G_3$$

$$B_2 = G_3 \oplus G_2$$

$$B_1 = G_3 \oplus G_2 \oplus G_1$$

$$B_0 = G_3 \oplus G_2 \oplus G_1 \oplus G_0$$

Procedure:

1. Verify the gates.
2. Make the connections as per the circuit diagram.
3. Switch on VCC and apply various combinations of input according to truth table.
4. For all input combinations the outputs are verified with the truth table.

		B1 B0			
		00	01	11	10
B3 B2	00				
	01	1	1	1	1
	11				
	10	1	1	1	1

G2:

		B1 B0			
		00	01	11	10
B3 B2	00				
	01				
	11	1	1	1	1
	10	1	1	1	1

K-Map for
K-Map for

TAL ELECTRONICS LAB

G3:

$$G2 = B2 \oplus B3$$

$$G3 = B3$$

K-Map for Gray to Binary Code Converter:

K-Map for B0:

G3 G2 \ G1 G0				
	00	01	11	10
00		1		1
01	1		1	
11		1		1
10	1		1	

$$B0 = G1 \oplus G2 \oplus G3 \oplus G4$$

K-Map for B1:

G3 G2 \ G1 G0				
	00	01	11	10
00			1	1
01	1	1		
11			1	1
10	1	1		

$$B1 = G1 \oplus G2 \oplus G3$$

K-Map for B2:

G3 G2 \ G1 G0				
	00	01	11	10
00				
01	1	1	1	1
11				
10	1	1	1	1

$$B2 = G2 \oplus G3$$

K-Map for B3:

G3 G2 \ G1 G0				
	00	01	11	10
00				
01				
11	1	1	1	1
10	1	1	1	1

$$B3 = G3$$

Result:

Thus Binary to Gray and Gray to Binary converters are designed, constructed using logic gates and their truth table was verified.

www.rejinpaul.com

www.rejinpaul.com

EX: NO:

DATE:

Design and implementation of 4 bit binary Adder/ Subtractor:

Aim:

To design and implement 4 bit binary adder and 4 bit binary subtractor using IC 7483.

Apparatus Required:

Sl. No.	COMPONENT	SPECIFICATION	QTY
1.	4 BIT BINARY ADDER	IC 7483	1
2.	X-OR GATE	IC 7486	1
3.	DIGITAL IC TRAINER KIT	-	1
4.	PATCH CORD	-	-

Theory:

Four bit binary adder:

A 4 bit binary adder can be constructed using four full adders. Here the full adders are connected in cascade, with the output carry from each full adder connected to the input carry of next full adder in chain. The input carry to the adder is C_1 and it ripples through the full adder to the output carry C_4 .

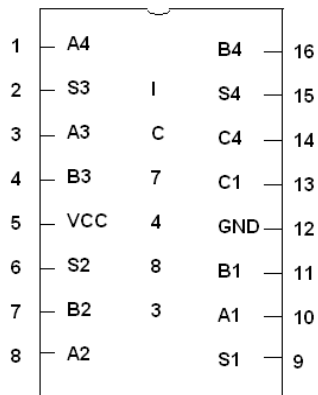
Four bit binary subtractor:

Binary subtraction is done using 2's complement subtraction method. For subtracting B from A using adders, $S = A + B' + 1$. A 4 bit binary subtractor using 4 bit binary adder consists of inverted B inputs and the carry input C_1 set to '1'.

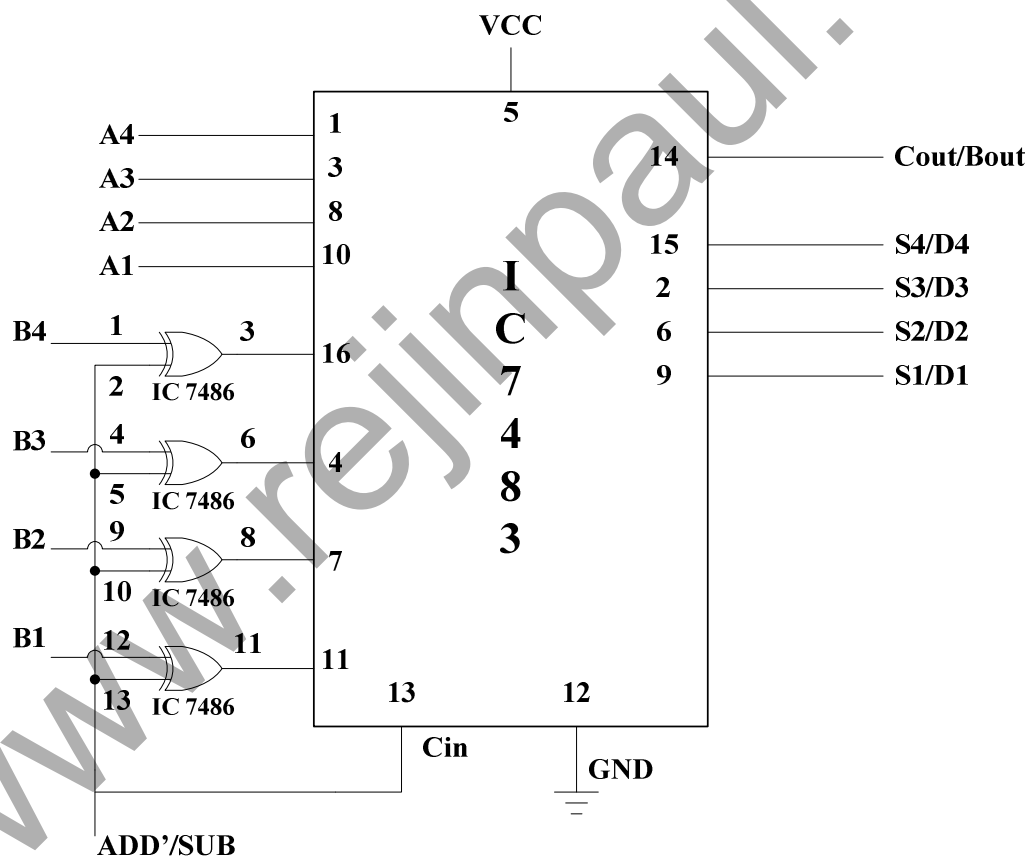
Four bit binary adder/subtractor:

A 4 bit binary adder/subtractor is used to perform both addition and sub-traction using a control line add'/sub. If add'/sub = 0 binary addition takes place and if add'/sub = 1 binary subtraction takes place, i.e. 2's complement subtraction of B from A.

Pin Diagram:



4 Bit Adder/Subtractor circuit diagram:



Procedure:

1. Verify the gates.
2. Make the connections as per the circuit diagram.
3. Switch on VCC and apply and verify various combinations of input according to the truth table for 4 bit binary adder/subtractor,
 - a) By keeping add'/sub as low, binary addition takes place.
 - b) By keeping add'/sub as high, binary subtraction takes place.

www.rejinpaul.com

Adder: Add'/sum = 0, Sub-tractor: Add'/Sum = 1

Input Data A				Input Data B				Addition					Subtraction				
A4	A3	A2	A1	B4	B3	B2	B1	Cout	S4	S3	S2	S1	Bout	D4	D3	D2	D1
1	0	0	0	0	0	1	0	0	1	0	1	0	1	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0	0	1	0	1	0	0	1	0	1	0
0	0	0	1	0	1	1	1	0	1	0	0	0	0	1	0	1	0
1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	0	1	0	1	1	1	1
1	0	1	0	1	1	0	1	1	0	1	1	1	0	1	1	0	1

Result:

Thus the 4 bit binary adder, 4 bit binary subtractor was designed and implement using IC 7483 with their truth table verified.

www.rejinpaul.com

www.rejinpaul.com

EX: NO:

DATE:

Design and Implementation of BCD Adder:

Aim:

To design and implement BCD adder using 4 bit binary adder IC 7483.

Apparatus Required:

Sl. No.	COMPONENT	SPECIFICATION	QTY
1.	4 BIT BINARY ADDER	IC 7483	2
2.	AND GATE	IC 7408	1
3.	OR GATE	IC 7432	1
4.	DIGITAL IC TRAINER KIT	-	1
5.	PATCH CORD	-	-

Theory:

BCD Addition:

Binary Coded Decimal is a method of using binary digits to represent the decimal digits 0 through 9. The valid BCD numbers are $(0000 \text{ to } 1001)_{\text{BCD}}$. Each digit of the decimal number will be represented by its four bit binary equivalent. Ex: $(127)_{10}$ - BCD equivalent $(0001 \ 0010 \ 0111)_2$. In BCD addition the following three cases are observed,

1. The resulting BCD number equal to less than $(1001)_{\text{BCD}}$.
2. The resulting BCD number greater than $(1001)_{\text{BCD}}$.
3. Carry is generated in the BCD addition.

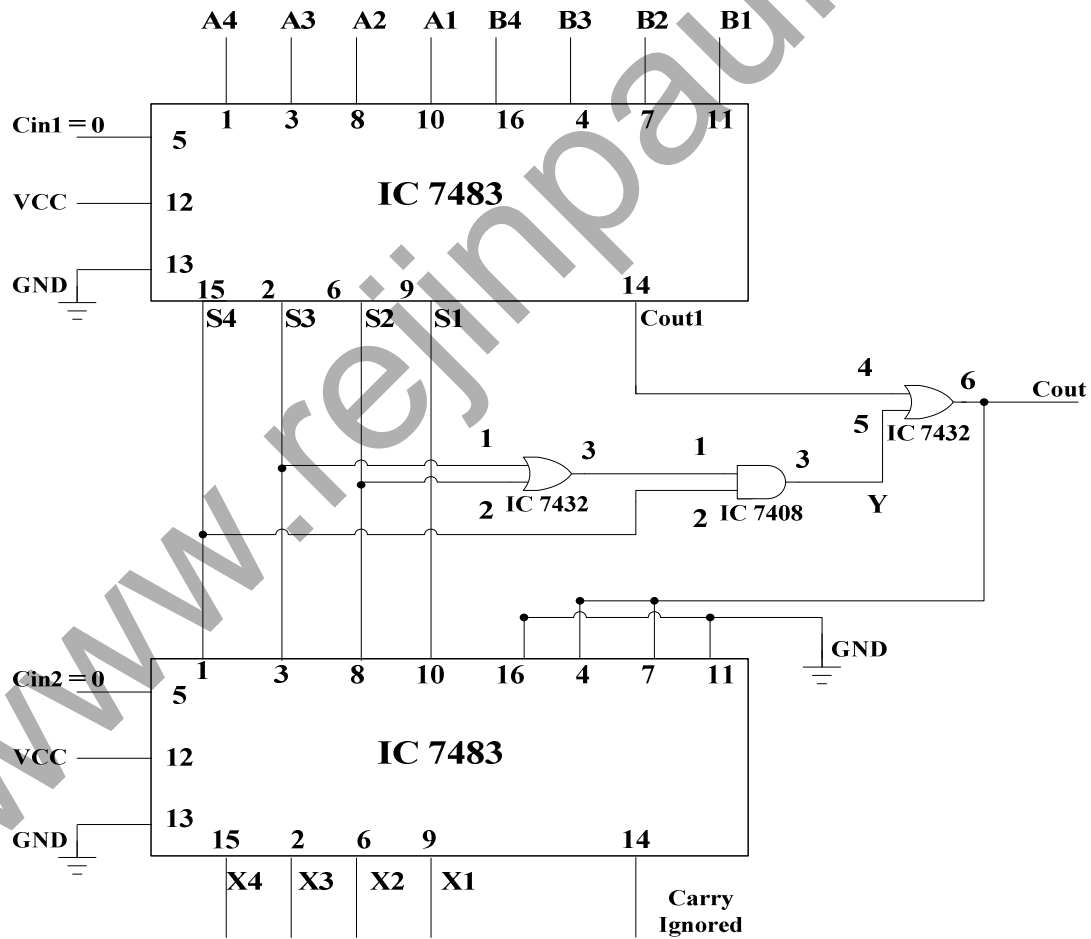
For case 2 and 3, the result is added with correction factor $(0110)_{\text{BCD}}$ so that the result is in valid BCD number.

Pin Diagram:

1	A4		B4	16
2	S3	I	S4	15
3	A3	C	C4	14
4	B3	7	C1	13
5	VCC	4	GND	12
6	S2	8	B1	11
7	B2	3	A1	10
8	A2		S1	9

Logic Diagram:

BCD Adder:



BCD Adder:

The two BCD inputs to be added are applied at inputs A and B of the first binary adder IC 7483. The sum output of the first binary adder is given to the B input of the second binary adder. The A input of the binary adder is given $(0110)_{BCD}$ when a carry is generated from the first adder or when sum from the first binary adder is greater than $(0110)_{BCD}$, else A input is $(0000)_{BCD}$. The following Boolean expression is used to find whether $(0110)_{BCD}$ or $(0000)_{BCD}$ needs to be applied to the A input,

$$\text{Cout} = \text{Cout1} + S_4 (S_3 + S_2)$$

Where S_4, S_3, S_2, S_1 are the sum of the BCD from the first binary adder with S_4 as the MSB and S_1 as the LSB. Cout1 is the carry output from the first binary adder.

Procedure:

1. Verify the gates.
2. Make the connections as per the circuit diagram.
3. Apply and verify the various combination of input according to the truth table for BCD adder.

BCD SUM				Output
S4	S3	S2	S1	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

S4 S3 \ S2 S1		S2 S1			
		00	01	11	10
S4 S3	00	0	0	0	0
	01	0	0	0	0
	11	1	1	1	1
	10	0	0	1	1

$$Y = S4 (S3 + S2)$$

Truth Table for BCD Adder:

Input Data A				Input Data B				BCD output				
A4	A3	A2	A1	B4	B3	B2	B1	Cout	X4	X3	X2	X1
1	0	0	0	0	0	1	0	1	0	0	0	0
1	0	0	0	1	0	0	0	1	0	1	1	0
1	0	0	1	1	0	0	1	1	1	0	0	0
0	1	1	1	0	0	0	1	0	1	0	0	0

Result:

Thus BCD adder was designed and implemented using IC 7483 with their truth table verified.

www.rejinpaul.com

www.rejinpaul.com

EX: NO:

DATE:

Design and Implementation of 2 Bit Magnitude Comparator:

Aim:

To design and implement of 2 bit Magnitude Comparator using logic gates

Apparatus Required:

SI. No.	COMPONENT	SPECIFICATION	QTY
1.	NOT GATE	IC 7404	2
2.	AND GATE	IC 7408	1
3.	OR GATE	IC 7432	1
4.	X-OR GATE	IC 7486	1
5.	3-Input AND Gate	IC 7411	2
6.	DIGITAL IC TRAINER KIT	-	1
7.	PATCH CORD	-	-

Theory:

The comparison of two numbers is an operation that determines one number is greater than, less than (or) equal to the other number. A magnitude comparator is a combinational circuit that compares two numbers A and B to determine their relative magnitude. The outcome of the comparator is specified by three binary variables that indicate whether $A > B$, $A = B$ (or) $A < B$. Consider two numbers A and B with two digits each. Here $A = A_1 A_0$ and $B = B_1 B_0$.

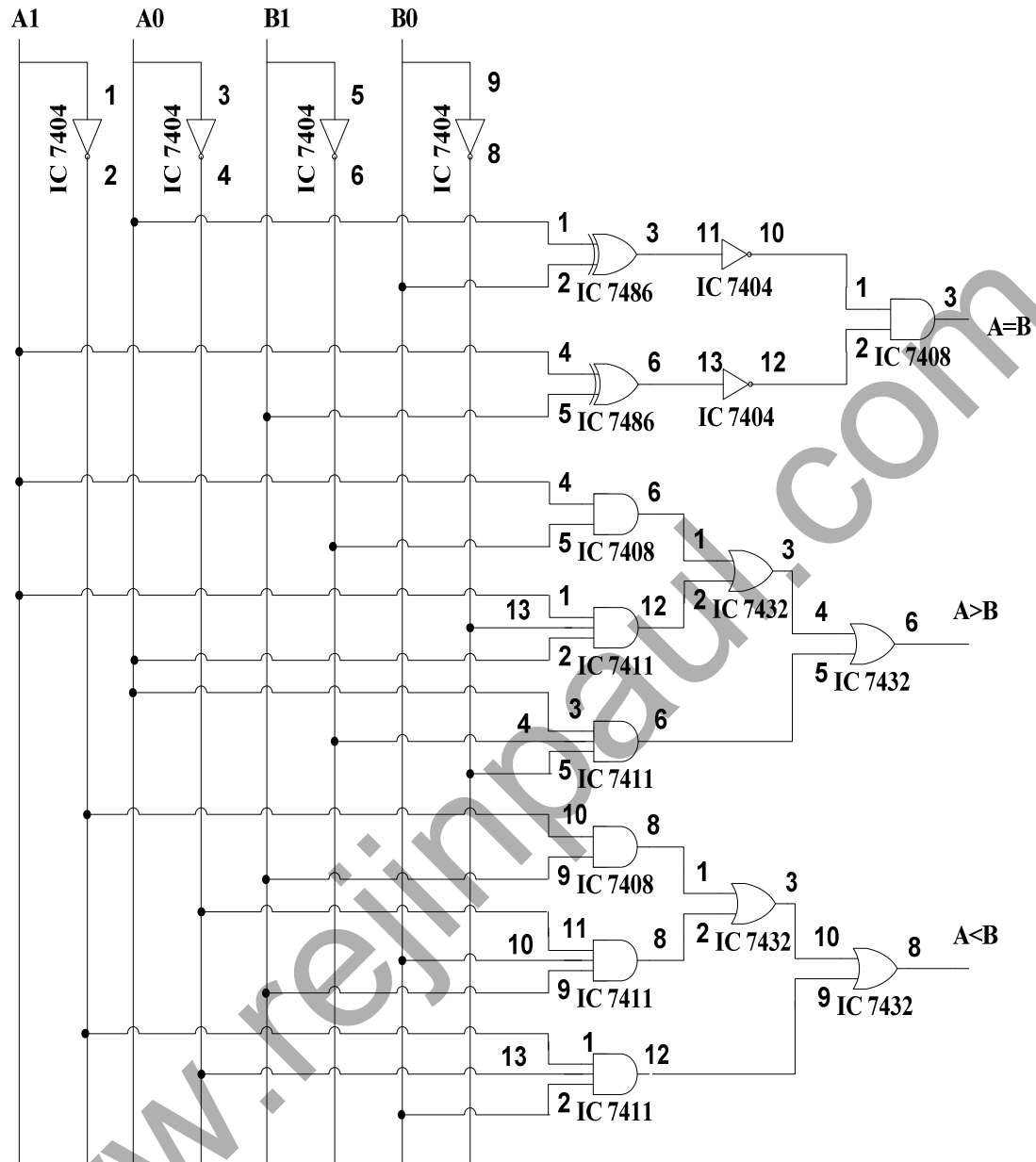
Now the Boolean equation for the two numbers to be equal to, greater than and lesser than are as follows,

$$(A=B) = (A_0 \odot B_0) (A_1 \odot B_1)$$

$$(A > B) = A_0 B_0' B_1' + A_1 B_1' + A_1 A_0 B_0'$$

$$(A < B) = A_1' A_0' B_0 + A_0' B_0 B_1 + A_1' B_1$$

Logic Diagram:**Two Bit Magnitude Comparator:**



Procedure:

1. Verify the gates.

2. Make the connections as per the circuit diagram.
3. Switch on VCC and apply various combinations of input according to truth table.
4. For all input combinations the outputs are verified with the truth table.

www.rejinpaul.com

Truth Table for Two Bit Magnitude Comparator:

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

K-Map for Two Bit Magnitude Comparator:

K-Map for (A>B):

A1 A0	B1 B0			
	00	01	11	10
00				
01	1			
11	1	1		1
10	1	1		

K-Map for (A<B):

A1 A0	B1 B0			
	00	01	11	10
00		1	1	1
01			1	1
11				
10			1	

$$[A > B] = A0B0'B1' + A1B1' + A1A0B0'$$

$$[A < B] = A1'A0'B0 + A0'B0B1 + A1'B1$$

K-Map for (A=B):

A1 A0	B1 B0			
	00	01	11	10
00	1			
01		1		
11			1	
10				1

$$[A=B] = (A0 \odot B0) (A1 \odot B1)$$

Result:

Thus the 2 bit Magnitude Comparator was designed and implemented using logic gates with their truth table verified.

www.rejinpaul.com

www.rejinpaul.com

EX: NO:

DATE:

Design and Implementation of 8 Bit Magnitude Comparator:

Aim:

To design and implement 8 bit Magnitude Comparator using IC 7485.

Apparatus Required:

SI. No.	COMPONENT	SPECIFICATION	QTY
1.	4-Bit Magnitude comparator	IC 7485	2
2.	DIGITAL IC TRAINER KIT	-	1
3.	PATCH CORD	-	-

Theory:

The magnitude comparator is a combinational circuit that compares two numbers A and B to determine their relative magnitude. The outcome of the comparator is specified by three binary variables that indicate whether $A > B$, $A = B$ (or) $A < B$.

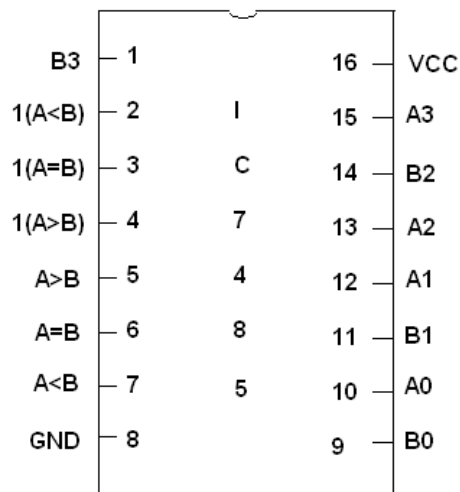
Eight Bit Magnitude Comparator using IC 7485:

IC 7485 is a four bit magnitude comparator that compares two four bit inputs (A_3, A_2, A_1, A_0) and (B_3, B_2, B_1, B_0). By cascading IC 7485 it is possible to construct n-bit comparators. Two IC 7485 is cascaded for the construction of one eight bit comparator. First the LSB of both A and B namely (A_3, A_2, A_1, A_0) and (B_3, B_2, B_1, B_0) is compared for cascading outputs $1(A > B)$, $1(A = B)$ and $1(A < B)$. The operation of the first four bit comparator is as follows. Depending on whether $A_3 > B_3$ (or) $A_3 < B_3$ the cascading outputs of the 4-bit comparator $1(A > B)$ (or) $1(A < B)$ are activated. But if $A_3 = B_3$, then the next MSB bits B_2 and A_2 are compared. Similarly, if $A_2 = B_2$ then comparison of A_1 and B_1 is performed and so on. Note if all the inputs (A_3, A_2, A_1, A_0) and (B_3, B_2, B_1, B_0) are equal then the IC 7485 will check for the cascading inputs. Function table for IC 7485 consolidates this operation. The intermediate outputs of the first four bit comparator are the intermediate inputs to the second four bit comparator. The second magnitude comparator gives the final outputs, whether $A > B$ (or) $A < B$ (or) $A = B$.

Procedure:

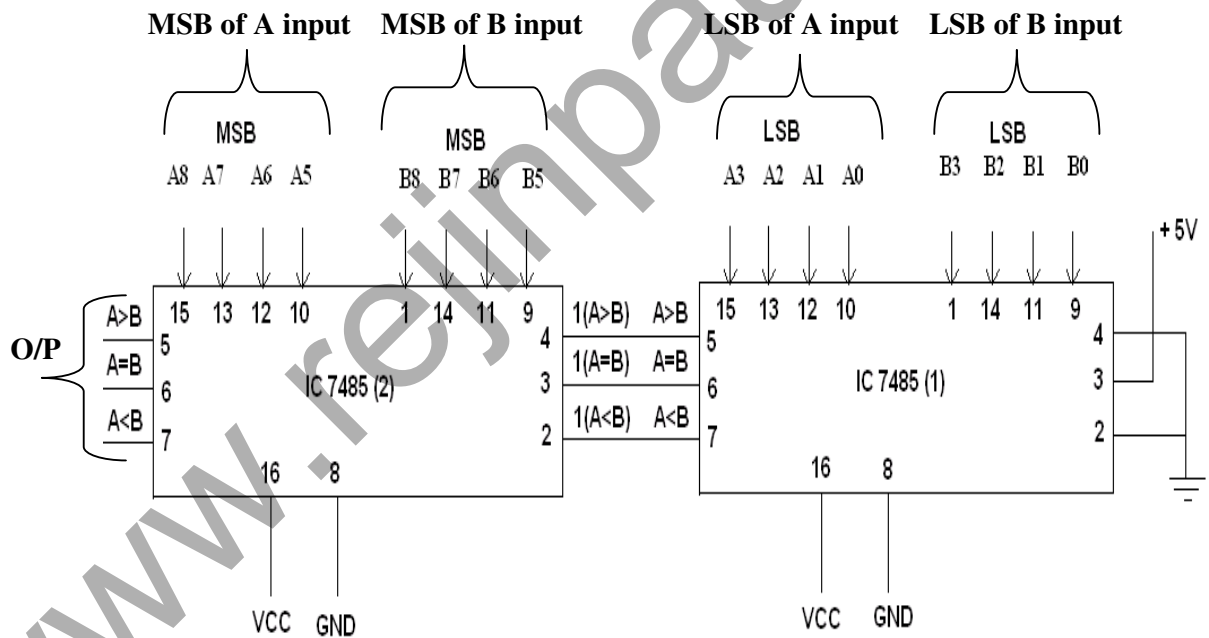
1. Verify the gates.
2. Make the connections as per the circuit diagram.
3. Switch on VCC and apply various combinations of input according to truth table.
4. For all input combinations the outputs are verified with the truth table.

Pin Diagram IC 7485:



Logic Diagram:

Eight Bit Comparator using Two – Four Bit Comparator:



www.rejinpaul.com

Function Table for IC 7485:

Comparing inputs				Cascading inputs			Outputs		
A ₃ ,B ₃	A ₂ ,B ₂	A ₁ ,B ₁	A ₀ ,B ₀	1(A>B)	1(A<B)	1(A=B)	A>B	A<B	A=B
A ₃ >B ₃	X	X	X	X	X	X	1	0	0
A ₃ <B ₃	X	X	X	X	X	X	0	1	0
A ₃ =B ₃	A ₂ >B ₂	X	X	X	X	X	1	0	0
A ₃ =B ₃	A ₂ <B ₂	X	X	X	X	X	0	1	0
A ₃ =B ₃	A ₂ =B ₂	A ₁ >B ₁	X	X	X	X	1	0	0
A ₃ =B ₃	A ₂ =B ₂	A ₁ <B ₁	X	X	X	X	0	1	0
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ >B ₀	X	X	X	1	0	0
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ <B ₀	X	X	X	0	1	0
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	1	0	0	1	0	0
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	0	1	0	0	1	0
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	0	0	0	1	1	0
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	X	X	1	0	0	1
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	1	1	0	0	0	0

TRUTH TABLE:

Inputs		Outputs		
[A8.....A0]	[B8.....B0]	A>B	A=B	A<B
1111 1111	1111 1111	0	1	0
1000 0000	0000 1000	1	0	0
0000 1000	1000 0000	0	0	1

Result:

Thus eight bit Magnitude Comparator was designed and implemented using IC 7485 with their truth table verified.

www.rejinpaul.com

www.rejinpaul.com

EX: NO:

DATE:

Design and Implementation of 16 Bit Odd/Even Parity Generator:

Aim:

To design and implement 16 bit odd/even parity generator using IC 74180.

Apparatus Required:

Sl. No.	COMPONENT	SPECIFICATION	QTY
1.	NOT Gate	IC 7404	1
2.	8-bit parity generator/ checker	IC 74180	2
3.	Digital IC trainer kit	-	1
4.	Patch cord	-	-

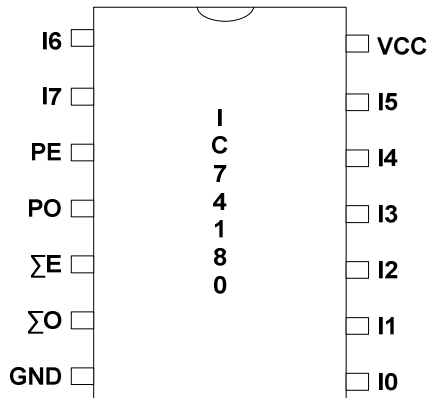
Theory:

When digital data is transmitted from one location to another, it is necessary to know at the receiving end, whether data received is free from errors. To help make the transmission accurate, special error detection methods are used. To detect errors there must be constant check on the data being transmitted. To check accuracy of the data an extra bit can be generated and transmitted along with the data. This bit is called the parity bit. A parity bit is used for detecting errors during transmission of binary information.

Parity generators are circuits that accept an n-1 bit data stream and generate an extra bit that is transmitted with the bit stream. This extra bit is referred to as parity bit. In an even parity bit scheme, the parity bit is '1' if there are odd number of 1's in the data stream and the parity bit is '0' if there are even number of 1's in the data stream. In the case of odd parity bit scheme, the reverse happens, that is the parity bit is '0' for odd number of 1's and '1' for even number of 1's in the bit stream.

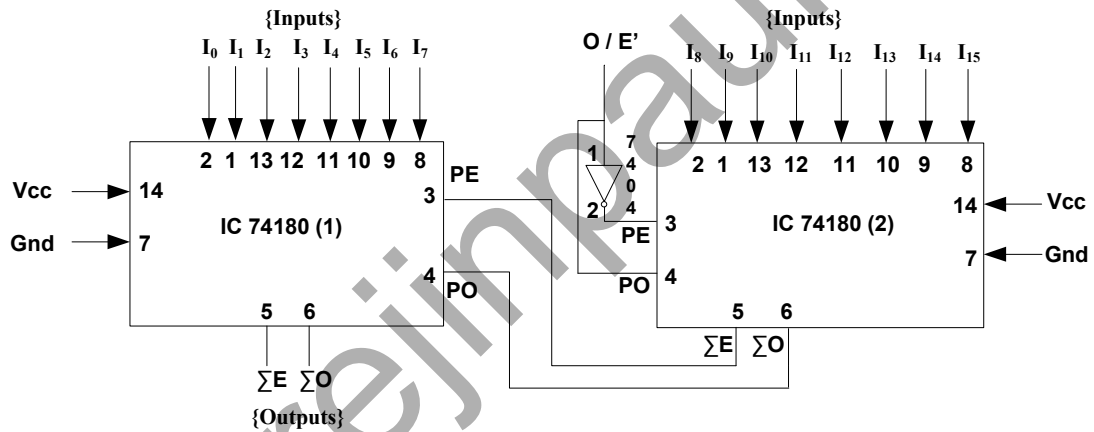
PIN DIAGRAM FOR IC 74180:

PIN DESCRIPTION:



Logic Diagram:

16 Bit Odd/Even Parity Generator:



Truth Table:

I0 I1 I2 I3 I4 I5 I6 I7	I8 I9 I10 I11 I12 I13 I14 I15	Active	ΣE	ΣO
1 1 0 0 0 0 0 0	1 1 0 0 0 0 0 0	1	1	0
1 1 0 0 0 0 0 0	1 1 0 0 0 0 0 0	0	0	1
1 1 0 0 0 0 0 0	0 1 0 0 0 0 0 0	0	1	0

Procedure:

1. Verify the gates.

2. Make the connections as per the circuit diagram.
3. Switch on VCC and apply various combinations of input according to truth table.
4. For all input combinations the outputs are verified with the truth table.

www.rejinpaul.com

Result:

Thus a 16-bit parity generator was designed and implemented using IC74180 with its truth table verified.

www.rejinpaul.com

EX: NO:

DATE:

Design and Implementation of 16 Bit Odd/Even Parity Checker:

Aim:

To design and implement 16 bit odd/even parity checker using IC 74180.

Apparatus Required:

Sl. No.	COMPONENT	SPECIFICATION	QTY
1.	NOT Gate	IC 7404	1
2.	8-bit parity generator/ checker	IC 74180	2
3.	Digital IC trainer kit	-	1
4.	Patch cord	-	-

Theory:

A parity bit is used for detecting errors during transmission of binary information. A parity bit is an extra bit included with a binary message to make the number as either even or odd. The message including the parity bit is transmitted and then checked at the receiver end for errors. An error is detected if the checked parity bit doesn't correspond to the transmitted parity bit. The circuit that generates the parity bit in the transmitter is called a 'parity generator' and the circuit that checks the parity in the receiver is called a 'parity checker'.

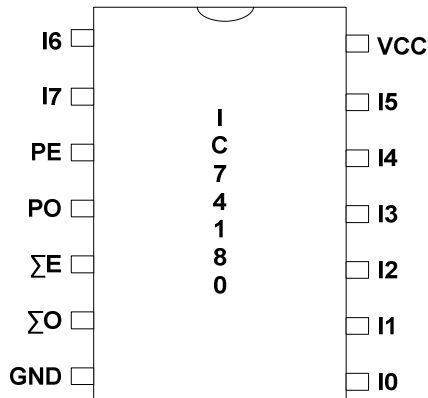
In even parity, the added parity bit will make the total number of 1's as even. In odd parity, the added parity bit will make the total number of 1's as odd. The parity checker circuit checks for possible errors in the transmission. If the information is passed in even parity, then the bits required must have an even number of 1's. An error occurs during transmission, if the received bits have an odd number of 1's indicating that one bit has changed in value during transmission.

PIN DIAGRAM FOR IC 74180:**PIN DESCRIPTION:**

$I_0 - I_7$ = Data Inputs

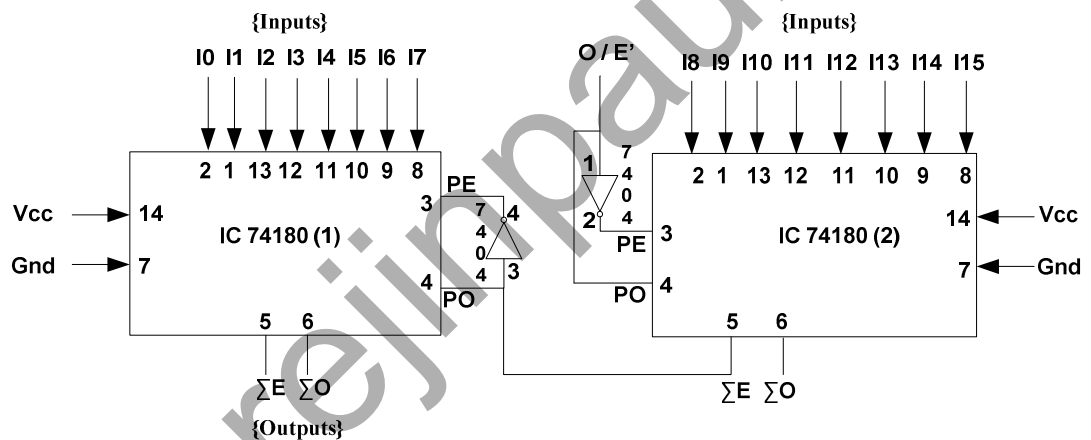
PO = Odd Input

PE = Even Input



Logic Diagram:

16 Bit Odd/Even Parity Checker:



Truth Table:

I7 I6 I5 I4 I3 I2 I1 I0	I7' I6' I5' I4' I3' I2' I1' I0'	Active	ΣE	ΣO
0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0	1	1	0
0 0 0 0 0 1 1 0	0 0 0 0 0 1 1 0	0	1	0
0 0 0 0 0 1 1 0	0 0 0 0 0 1 1 0	1	0	1

Procedure:

1. Verify the gates.

2. Make the connections as per the circuit diagram.
3. Switch on VCC and apply various combinations of input according to truth table.
4. For all input combinations the outputs are verified with the truth table.

www.rejinpaul.com

Result:

Thus a 16-bit parity checker was designed and implemented using IC74180 with its truth table verified.

www.rejinpaul.com

EX: NO:

DATE:

Design and Implementation of Multiplexer and De-multiplexer:

Aim:

To design and implement multiplexer and de-multiplexer using logic gates, study of IC 74150 and IC 74154.

Apparatus Required:

Sl. No.	COMPONENT	SPECIFICATION	QTY
1.	NOT GATE	IC 7404	1
2.	OR GATE	IC 7432	1
3.	3-Input AND Gate	IC 7411	2
4.	DIGITAL IC TRAINER KIT	-	1
5.	PATCH CORD	-	-

Theory:**Multiplexer:**

Multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line. Selection of input lines controlled by a set of selection lines – 2^n input lines and n selection lines. Multiplexers are used to form a selected path between multiple sources and single destination.

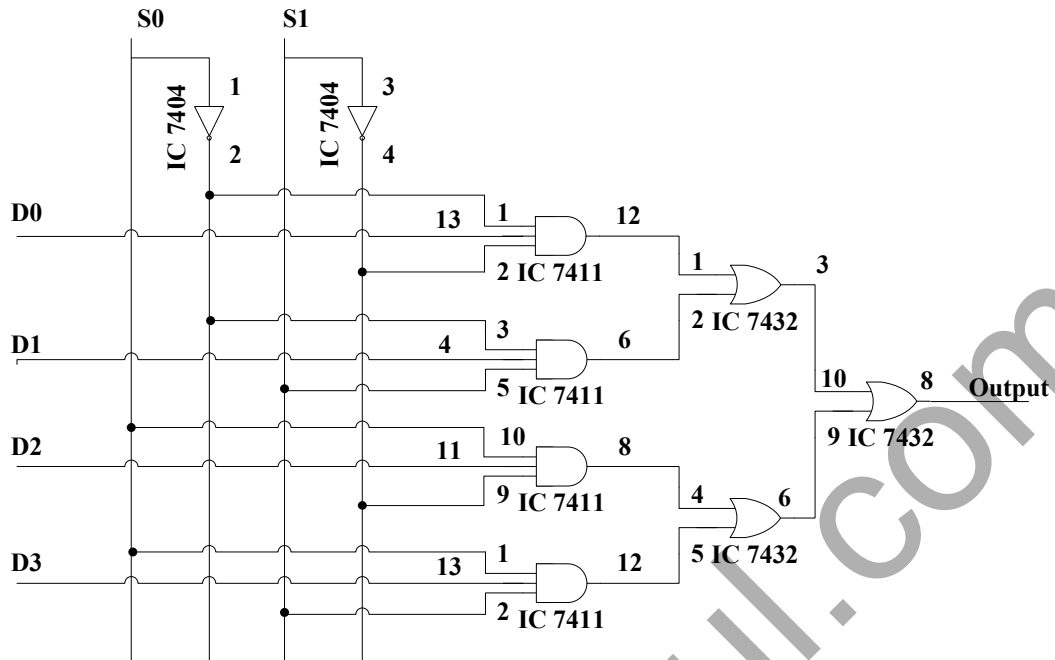
De-Multiplexer:

De-multiplexer is a circuit that receives information on single line and transmits the information on one of 2^n possible output lines. Selection of specific output lines is controlled by n selection lines.

IC74150 – 16 to 1 Multiplexer:

IC74150 is a data selector/multiplexer IC that selects one of the sixteen data sources E0 to E15. The STROBE input of the IC must be at a low logic level to enable these devices. A high level at the STROBE forces the Q output high. IC 74150 is used in parallel to serial conversion. In digital communication, a number of input lines are connected to a single output channel using multiplexer, so that information transmitted one by one in a time shared basis.

Logic Diagram:**4:1 Multiplexer:**



Truth Table:

S0	S1	Output
0	0	D0
0	1	D1
1	0	D2
1	1	D3

$$\text{Output} = D_0 S_0' S_1' + D_1 S_0' S_1 + D_2 S_0 S_1' + D_3 S_0 S_1$$

IC74154 – 1 to 16 De-Multiplexer:

IC74154 will act as a 1 to 16 De-Multiplexer by using the 4 address lines A, B, C, and D to address the active low output lines Q0 to Q15, passing data from one of the strobe inputs

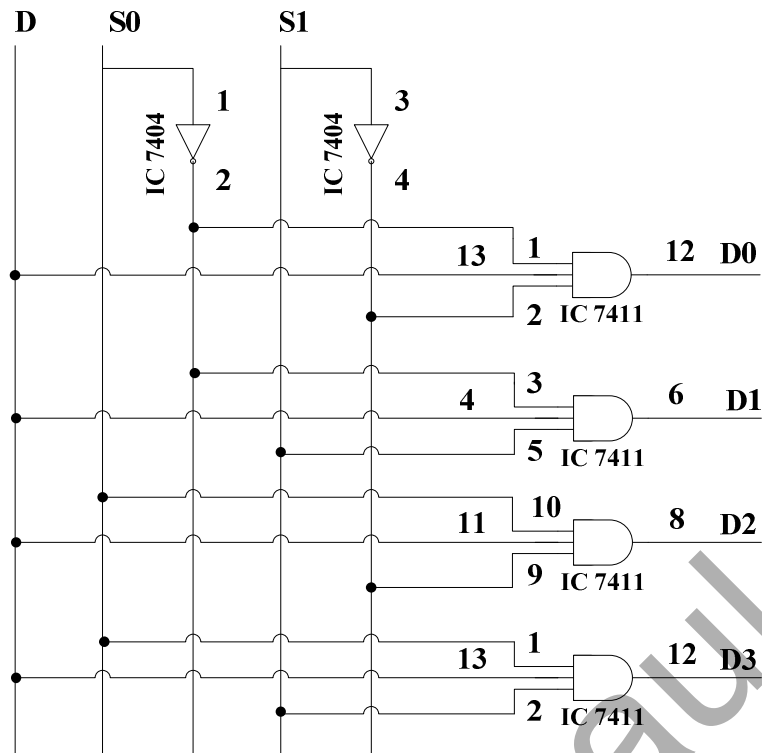
(FE1/FE2) with the other strobe input low. When either strobe inputs are high all the outputs are high. IC 74154 is ideally suited for implementing high-performance memory decoders. In telecommunication IC 74154 accepts a single input signal that carries many channels and separates those over multiple output signals.

Procedure:

1. Verify the gates.
2. Make the connections as per the circuit diagram.
3. Switch on VCC and apply various combinations of input according to truth table.
4. For all input combinations the outputs are verified with the truth table.

Logic Diagram:

1:4 De-multiplexer:



Truth Table:

Input			Output			
D	S0	S1	D0	D1	D2	D3
1	0	0	D	0	0	0
1	0	1	0	D	0	0
1	1	0	0	0	D	0
1	1	1	0	0	0	D

$$D0 = DS0S1'$$

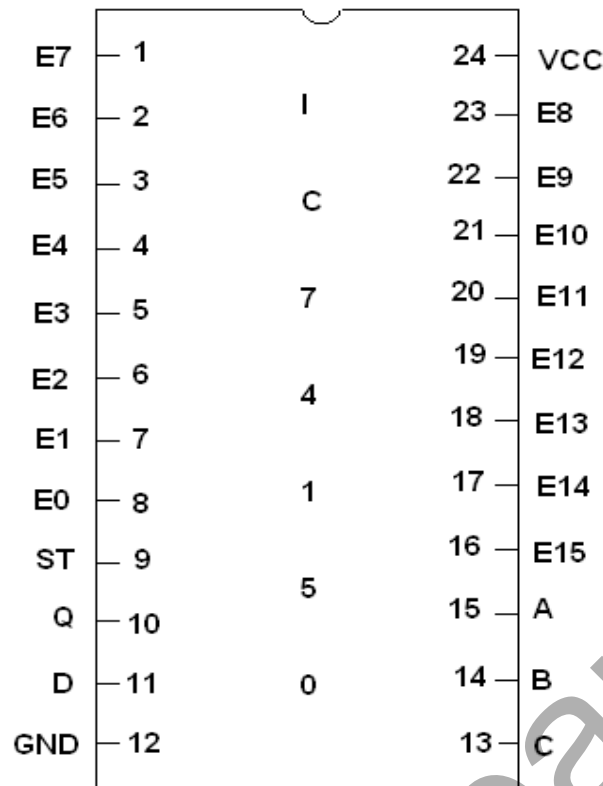
$$D1 = DS0'S1$$

$$D2 = DS0S1'$$

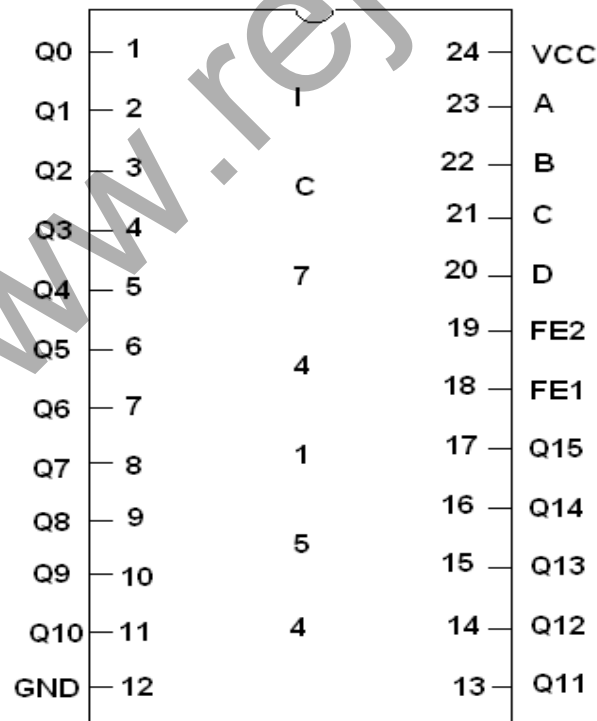
$$D3 = DS0S1$$

www.rejinpaul.com

Pin Diagram for IC 74150:



Pin Diagram for IC 74154:



Result:

Thus multiplexer and de-multiplexer was designed and implemented using logic gates with their truth table verified. Multiplexer and De-multiplexer, IC 74150 and IC 74154 are studied.

www.rejinpaul.com

www.rejinpaul.com

EX: NO:

DATE:

Design and Implementation of Encoder and Decoder:

Aim:

To design and implement encoder and decoder using logic gates, study of IC7445 and IC74147.

Apparatus Required:

Sl. No.	COMPONENT	SPECIFICATION	QTY
1.	NOT GATE	IC 7404	1
2.	OR GATE	IC 7432	1
3.	AND Gate	IC 7408	1
4.	DIGITAL IC TRAINER KIT	-	1
5.	PATCH CORD	-	-

Theory:

Encoder:

An encoder is a combinational logic circuit that has 2^n input lines and n output lines. As an example consider an four input and two output encoder. It is assumed that only one input has '1' at any given time. From truth table, it is obvious that the output is '1' for A when the input is 2 and 3; B is '1' when the input is 1 and 2.

De-Coder:

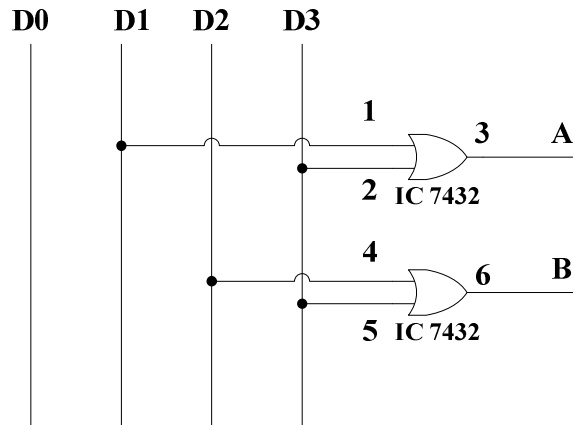
A decoder is a combinational circuit that converts n -bit binary input lines into 2^n output lines such that output line will be activated for only one of possible combination of inputs. The outputs are selected based on two select inputs. The inputs AB are decoded into four digits output each representing one of minterms of two input variables.

IC 7445 BCD to Decimal Decoder:

IC 7445 is a BCD to decimal decoder that accepts 4 active high inputs and produces ten active low outputs. Full decoding of BCD input logic ensures that all outputs remain OFF for all invalid (1010–1111) binary input conditions. IC 7445 is widely used in BCD to seven segment display. It is also used as address decoders where the decoded output is used as chip select signal to select the chip of interest.

Logic Diagram:

4x2 Encoder:



Truth Table:

Input				Output	
D0	D1	D2	D3	A	B
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

$$A = D2 + D3$$

$$B = D1 + D3$$

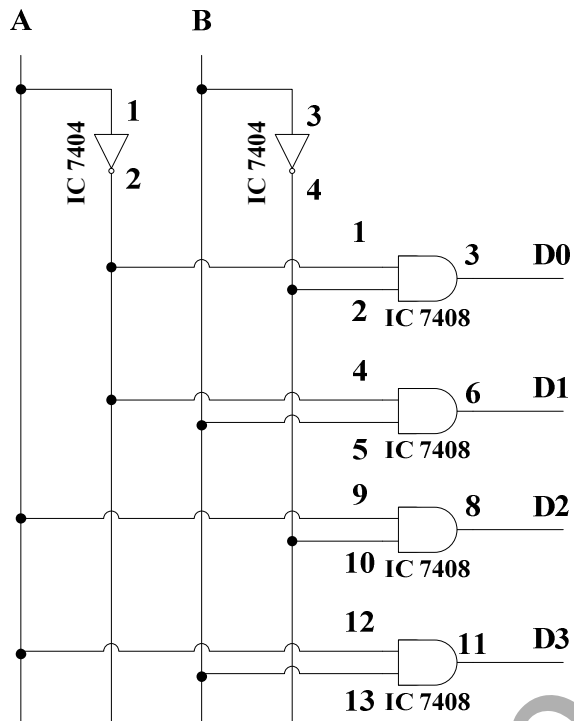
IC 74147 - Priority Encoder:

IC 74147 is a 9-input priority encoders that accepts data from nine active LOW inputs (E1 to E9) and provide a binary representation on the four active LOW outputs (QA to QD). A priority is assigned to each input so that when two or more inputs are simultaneously active, the input with the highest priority is represented on the output, with input line E9 having the highest priority. The devices provide the 10-line to 4-line priority encoding function by use of the implied decimal “zero”. The “zero” is encoded when all nine data inputs are HIGH, forcing all four outputs HIGH. IC 74147 encodes 10-line decimal to 4-line BCD. Used in 10-position switch encoding and also in code converters and generators.

Procedure:

1. Verify the gates.
2. Make the connections as per the circuit diagram.
3. Switch on VCC and apply various combinations of input according to truth table.
4. For all input combinations the outputs are verified with the truth table.

2x4 De-Coder:



Truth Table:

Input		Output			
A	B	D0	D1	D2	D3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$$D0 = A'B'$$

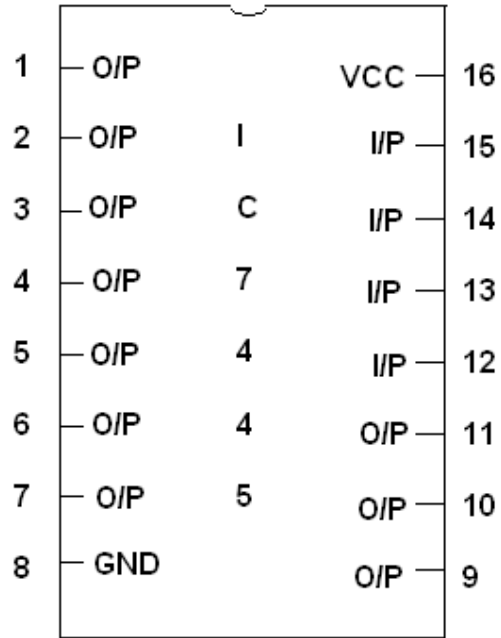
$$D1 = A'B$$

$$D2 = AB'$$

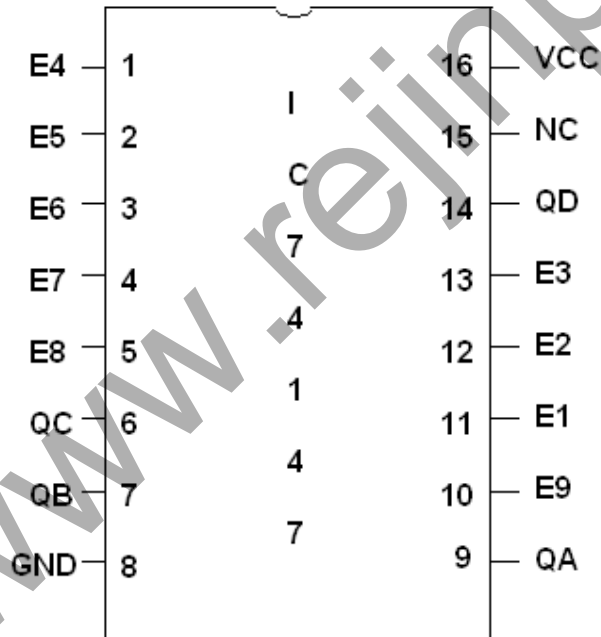
$$D3 = AB$$

www.rejinpaul.com

Pin Diagram IC7445:



Pin Diagram IC 74147:



Result:

Thus the Encoder and De-Coder were designed and implemented using logic gates with their truth table verified. Encoder and De-Coder, IC 7445 and IC 74147 are studied.

www.rejinpaul.com

www.rejinpaul.com

EX: NO:

DATE:

Construction and Verification of 4 Bit Ripple Counter:

Aim:

To construct and verify 4 bit Ripple Counter.

Apparatus Required:

SI. No.	COMPONENT	SPECIFICATION	QTY
1.	JK FLIP FLOP	IC 7473	2
2.	NAND GATE	IC 7400	1
3.	DIGITAL IC TRAINER KIT	-	1
4.	PATCH CORD	-	-

Theory:

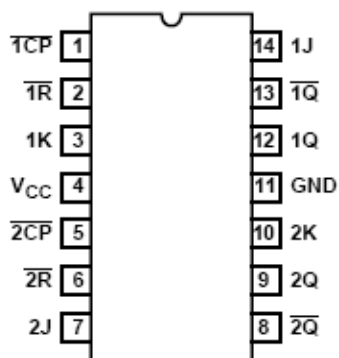
Counters are a group of flip flops connected together to perform counting operation. According to the way the flip flops are clocked, there are two types of flip flops,

- Asynchronous Counter
- Synchronous Counter

In asynchronous counter, the first flip flop is clocked by the external clock pulse. Then each successive flip flops are clocked by Q (or) Q' output of the previous flip flop. In 4-bit ripple counter, the total number of states is 16 (2^4) and this varies from 0000_2 to 1111_2

Procedure:

- Verify the flip flop.
- Make the connections as per the circuit diagram.
- Switch on VCC and apply various combinations of input according to truth table.
- By applying clock pulse, all the input combinations are given and the outputs are verified with the truth table.



Pin Diagram IC 7473:

Pin Description:

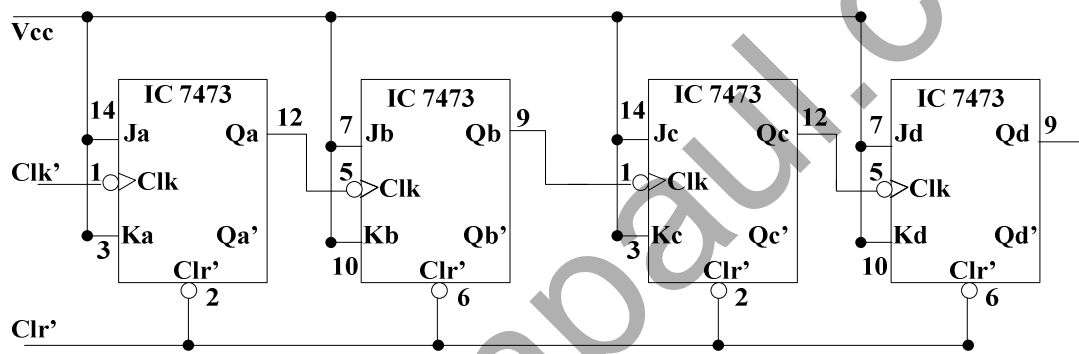
1CP', 2CP' = neg-edge clock input for JK Flip F

1R', 2R' = Negative Clear Input for JK Flip F

1J, 2J = J Input

Logic Diagram:

Four Bit Ripple Counter:



Truth Table for Four Bit Ripple Counter:

Clk	Qd	Qc	Qb	Qa
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Result:

Thus 4 bit Ripple Counter was constructed and verified with their truth table.

www.rejinpaul.com

www.rejinpaul.com

EX: NO:

DATE:

Construction and Verification of Mod-10 and Mod-12 Ripple Counters:

Aim:

To construct and verify Mod-10 and Mod-12 Ripple Counters.

Apparatus Required:

SI. No.	COMPONENT	SPECIFICATION	QTY
1.	JK FLIP FLOP	IC 7473	2
2.	NAND GATE	IC 7400	1
3.	DIGITAL IC TRAINER KIT	-	1
4.	PATCH CORD	-	-

Theory:

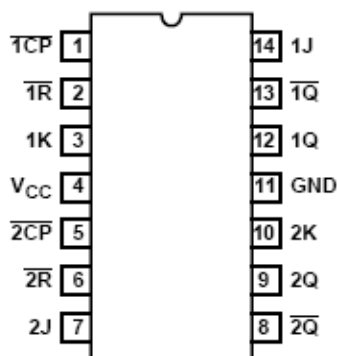
Counters are a group of flip flops connected together to perform counting operation. According to the way the flip flops are clocked, there are two types of flip flops,

- c) Asynchronous Counter
- d) Synchronous Counter

In asynchronous counter, the first flip flop is clocked by the external clock pulse. Then each successive flip flops are clocked by Q (or) Q' output of the previous flip flop. In 4-bit ripple counter, the total number of states is 16 (2^4) and this varies from 0000_2 to 1111_2 . If the counters are designed with number of sequence which is less than 2^n , then those counters are said to be Mod – N counters where N denotes number of sequence. Thus in Mod–10 counter, total number of states is 10 and number of flip flops are 4. Similarly in Mod–12 counter, total number of states is 10 and number of flip flops are 4.

Procedure:

1. Verify the flip flop.
2. Make the connections as per the circuit diagram.
3. Switch on VCC and apply various combinations of input according to truth table.
4. By applying the clock pulse, all the input combinations are given and the outputs are verified with the truth table.



Pin Diagram IC 7473:

~ 84 ~

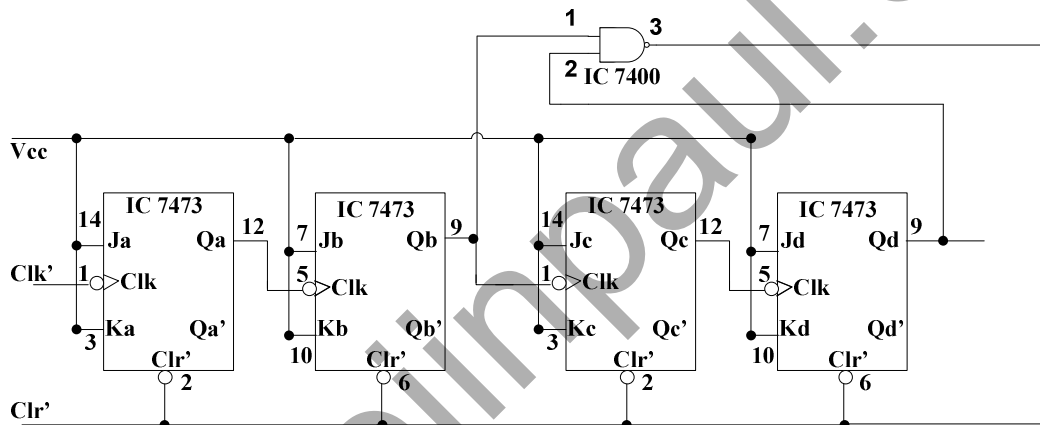
Pin Description:

1CP', 2CP' = edge clock input for JK Flip F

1R', 2R' = Negative Clear Input for JK Flip F

Logic Diagram:

Mod – 10 Counter:



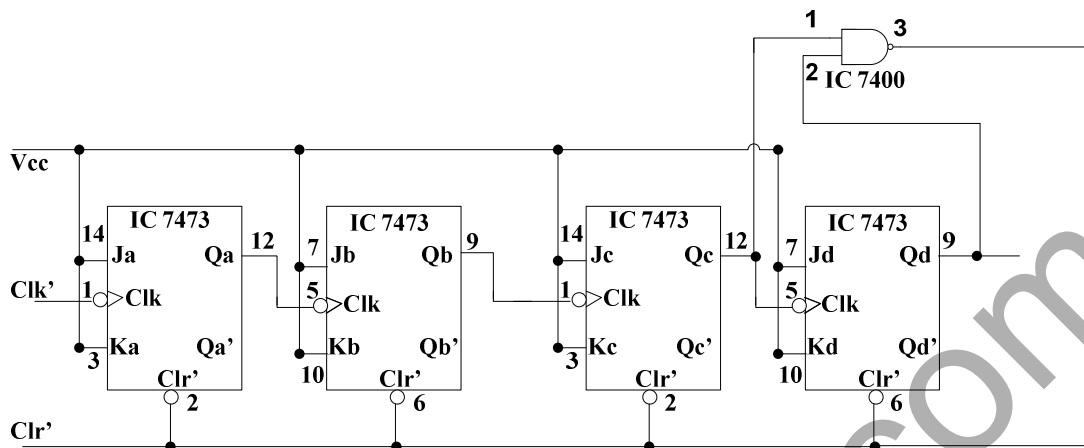
Truth Table for Mod – 10 Counter:

Clk	Qd	Qc	Qb	Qa
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	0	0	0	0

www.rejinpaul.com

Logic Diagram:

Mod – 12 Counter:



Truth Table for Mod – 12 Counter:

Clk	Qd	Qc	Qb	Qa
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	0	0	0	0

Result:

Thus Mod-10 and Mod-12 Ripple Counters was constructed and verified with their truth table.

www.rejinpaul.com

www.rejinpaul.com

EX: NO:

DATE:

Design and Implementation of 3 Bit Synchronous Up/Down Counter:

Aim:

To design, implement 3 bit synchronous up/ down counter and verify its truth table.

Apparatus Required:

SI. No.	COMPONENT	SPECIFICATION	QTY
1.	JK FLIP FLOP	IC 7473	2
2.	NOT GATE	IC 7404	1
3.	OR GATE	IC 7432	1
4.	AND Gate	IC 7408	1
5.	DIGITAL IC TRAINER KIT	-	1
6.	PATCH CORD	-	-

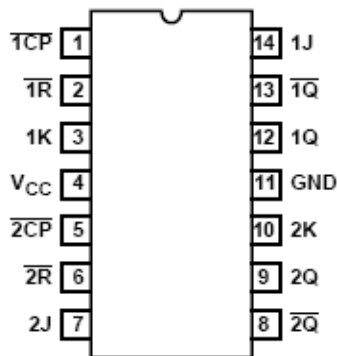
Theory:

A counter that advances upward through its sequence (0, 1, 2, 3...0, 1....) is called up counter. A counter that decrement downward through its sequence (3, 2, 1, 0, 3, 2) is called down counter. A up/down counter is a counter used to perform both up counting and down counting operation using up/down control signal.

Procedure:

1. Verify the flip flop.
2. Make the connections as per the circuit diagram.
3. Switch on VCC and apply various combinations of input according to truth table.
4. By applying the clock pulse, all input combinations are given and the outputs are verified with the truth table.

Pin Diagram IC 7473:



Pin Description:

1CP', 2CP' = neg-edge clock input for JK Flip Flop 1 and 2

1R', 2R' = Negative Clear Input for JK Flip Flop 1 and 2

1J, 2J = J Input

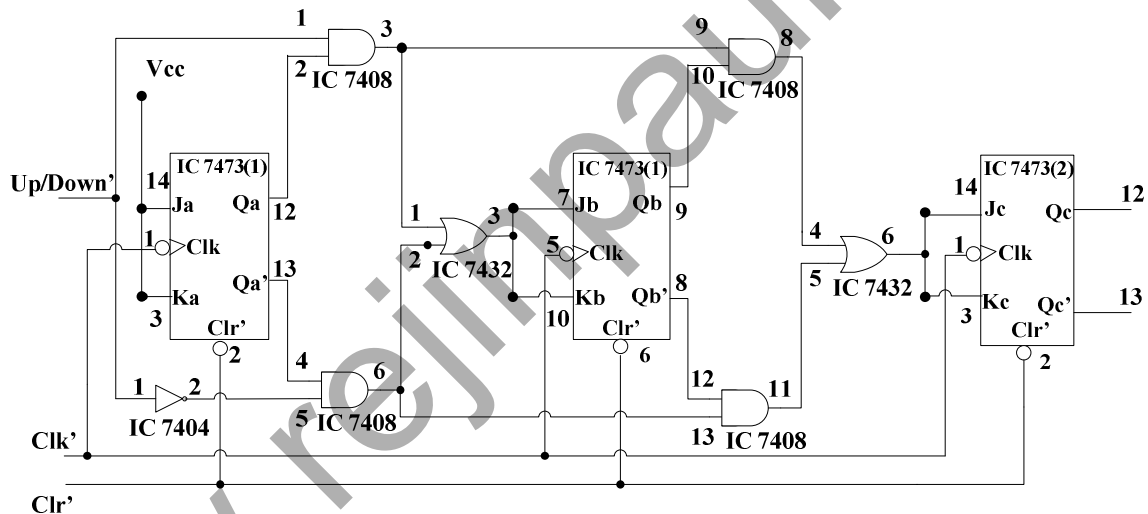
1k, 2k = K Input

1Q, 2Q = Q Output

1Q', 2Q' = Q' Output

Logic Diagram:

3-bit synchronous up/down counter:



Truth Table - JK Flip Flop:

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	Q_n'

Excitation Table - JK Flip Flop:

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

www.rejinpaul.com

Truth Table:

Input Up/Down'	Present State			Next State			Flip Flop Input					
X	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
1	0	0	0	0	0	1	0	X	0	X	1	X
	0	0	1	0	1	0	0	X	1	X	X	1
	0	1	0	0	1	1	0	X	X	0	1	X
	0	1	1	1	0	0	1	X	X	1	X	1
	1	0	0	1	0	1	X	0	0	X	1	X
	1	0	1	1	1	0	X	0	1	X	X	1
	1	1	0	1	1	1	X	0	X	0	1	X
	1	1	1	0	0	0	X	1	X	1	X	1
0	1	1	0	1	0	1	X	0	X	1	1	X
	1	0	1	1	0	0	X	0	0	X	X	1
	1	0	0	0	1	1	X	1	1	X	1	X
	0	1	1	0	1	0	0	X	X	0	X	1
	0	1	0	0	0	1	0	X	X	1	1	X
	0	0	1	0	0	0	0	X	0	X	X	1
	0	0	0	1	1	1	1	X	1	X	1	X
	1	1	1	1	1	0	X	0	X	0	X	1

K-Map Simplification:

K-Map for J_A:

XQC \ QBQA					
		00	01	11	10
00	00	1	X	X	1
01	01	1	X	X	1
11	11	1	X	X	1
10	10	1	X	X	1

$$J_A = 1$$

K-Map for J_B:

XQC \ QBQA					
		00	01	11	10
00	00	1	0	X	X
01	01	1	0	X	X
11	11	0	1	X	X
10	10	0	1	X	X

$$J_B = XQ_A + X'Q_A'$$

www.rejinpaul.com

K-Map for J_C :

		QBQA			
		00	01	11	10
XQC	00	1	0	0	0
	01	X	X	X	X
	11	X	X	X	X
	10	0	0	1	0

$$J_C = XQ_B' Q_A' + XQ_B Q_A$$

K-Map for K_A :

		QAQB			
		00	01	11	10
XQC	00	X	1	1	X
	01	X	1	1	X
	11	X	1	1	X
	10	X	1	1	X

$$K_A = 1$$

K-Map for K_B :

		QBQA			
		00	01	11	10
XQC	00	X	X	0	1
	01	X	X	0	1
	11	X	X	1	0
	10	X	X	1	0

$$K_B = X'Q_A' + XQ_A$$

K-Map for K_C :

		QBQA			
		00	01	11	10
XQC	00	X	X	X	X
	01	1	0	0	0
	11	0	0	1	0
	10	X	X	X	X

$$K_C = XQ_B' Q_A' + XQ_B Q_A$$

Result:

Thus 3 bit synchronous up/ down counter was designed and implemented with its truth table verified.

www.rejinpaul.com

www.rejinpaul.com

EX: NO:

DATE:

Implementation of SISO and SIPO Shift Registers:

Aim:

To implement SISO and SIPO shift registers using flip flop with its truth table verified.

Apparatus Required:

Sl. No.	COMPONENT	SPECIFICATION	QTY
1.	D FLIP FLOP	IC 7474	2
2.	DIGITAL IC TRAINER KIT	-	1
3.	PATCH CORD	-	-

Theory:

A register capable of shifting its binary information either to the right or to the left is called a shift register. The logic types of shift registers in terms of data movement are,

Serial In Serial Out:

SISO shift register accepts data serially (i.e.) one bit at a time on a single line and produces the stored data on its output also in serial form.

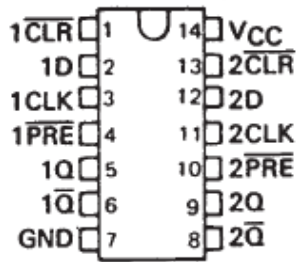
Serial In Parallel Out:

SIPO shift register accepts data serially and produces the stored information on its output in parallel form.

Procedure:

1. Verify the flip flop.
2. Make the connections as per the circuit diagram.
3. Switch on VCC and apply various combinations of input according to truth table.
4. By applying the clock pulse, all input combinations are given and the outputs are verified with the truth table.

Pin Diagram:

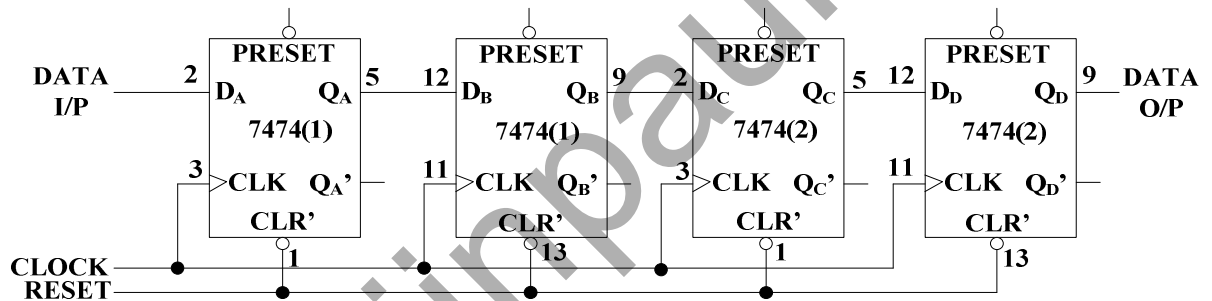


Pin Description:

- 1CLR', 2CLR' – Low level clear inputs.
- 1PRE', 2PRE' – Low level pre-set inputs.
- Vcc, Gnd – Power and Ground pins.
- 1CLK, 2CLK – Positive edge triggered clock inputs.
- 1D, 2D – Data inputs.
- 1Q, 2Q – Data outputs.
- 1Q', 2Q' – Complemented Data outputs.

Logic Diagram:

Serial In Serial Out:



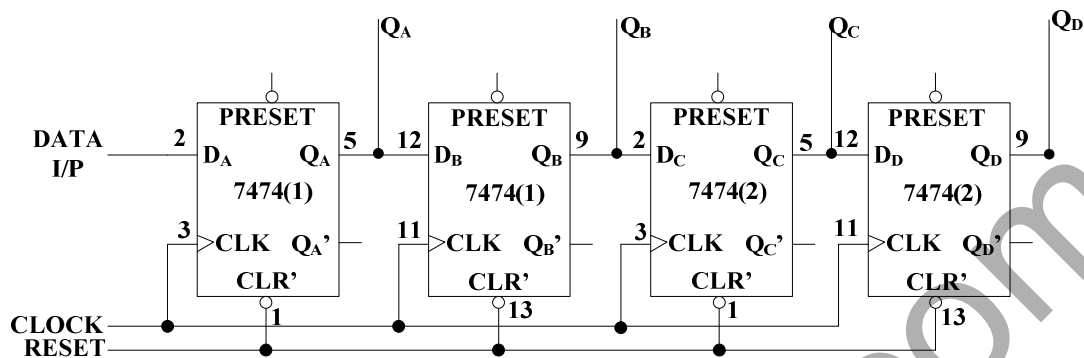
Truth Table for SISO Shift Register :

Clk	Serial input	Q _A	Q _B	Q _C	Q _D	Serial Output
0	1	0	0	0	0	0
1	1	1	0	0	0	0
2	0	1	1	0	0	0
3	1	0	1	1	0	0
4	0	1	0	1	1	1
5	0	0	1	0	1	1
6	0	0	0	1	0	0
7	0	0	0	0	1	1
8	0	0	0	0	0	0

www.rejinpaul.com

Logic Diagram:

Serial In Parallel Out:



Truth Table for SIPO Shift Register :

Clk	Serial input	Q _A	Q _B	Q _C	Q _D	Parallel Output [Q _A ... Q _D]
0	1	0	0	0	0	0000
1	1	1	0	0	0	1000
2	0	1	1	0	0	1100
3	1	0	1	1	0	0110
4	0	1	0	1	1	1011

Result:

Thus SISO and SIPO shift registers are implemented using flip flop with its truth table verified.

www.rejinpaul.com

www.rejinpaul.com

EX: NO:

DATE:

Implementation of PIPO and PISO Shift Registers:

Aim:

To implement PIPO and PISO shift registers using flip flop with its truth table verified.

Apparatus Required:

Sl. No.	COMPONENT	SPECIFICATION	QTY
1.	D FLIP FLOP	IC 7474	2
2.	NOT GATE	IC 7404	1
3.	OR GATE	IC 7432	1
4.	AND Gate	IC 7408	2
5.	DIGITAL IC TRAINER KIT	-	1
6.	PATCH CORD	-	-

Theory:

A register capable of shifting its binary information either to the right or to the left is called a shift register. The logic types of shift registers in terms of data movement are,

Parallel In Parallel Out:

PIPO shift register accepts data in parallel and produces the stored data on its output also in parallel form on application of the clock input.

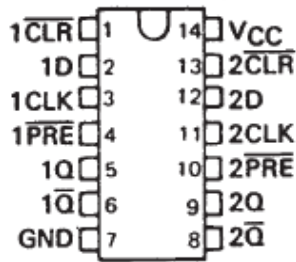
Parallel In Serial Out:

PISO shift register accepts data in parallel and produces the stored data on its output in serial form. The data are presented on the input lines D_a to D_d in parallel. The data loads into the register when the Shift/Load' control lines is held low. When the Shift/Load' is held high the data stored are clocked out serially.

Procedure:

1. Verify the flip flop.
2. Make the connections as per the circuit diagram.
3. Switch on VCC and apply various combinations of input according to truth table.
4. By applying the clock pulse, all input combinations are given and the outputs are verified with the truth table.

Pin Diagram:

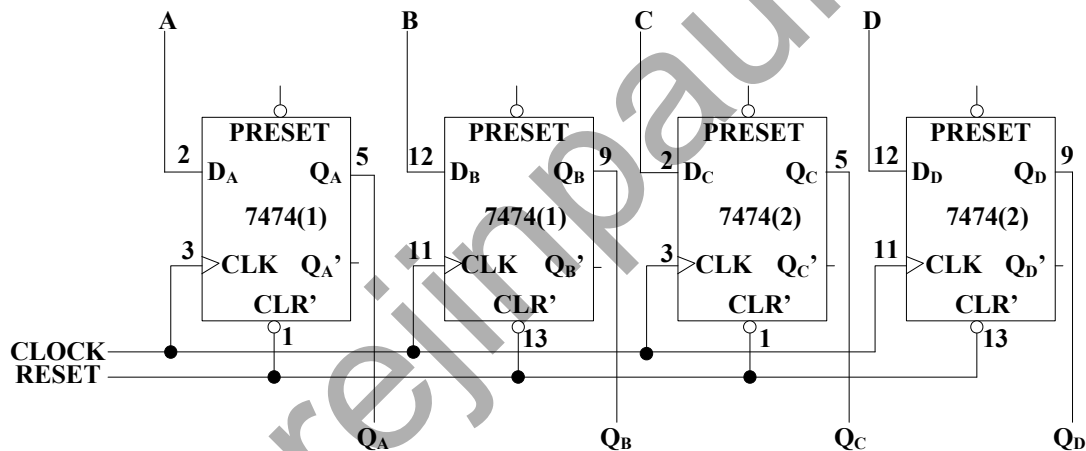


Pin Description:

- 1CLR', 2CLR' – Low level clear inputs.
- 1PRE', 2PRE' – Low level pre-set inputs.
- Vcc, Gnd – Power and Ground pins.
- 1CLK, 2CLK – Positive edge triggered clock inputs.
- 1D, 2D – Data inputs.
- 1Q, 2Q – Data outputs.
- 1Q', 2Q' – Complemented Data outputs.

Logic Diagram:

Parallel In Parallel Out:



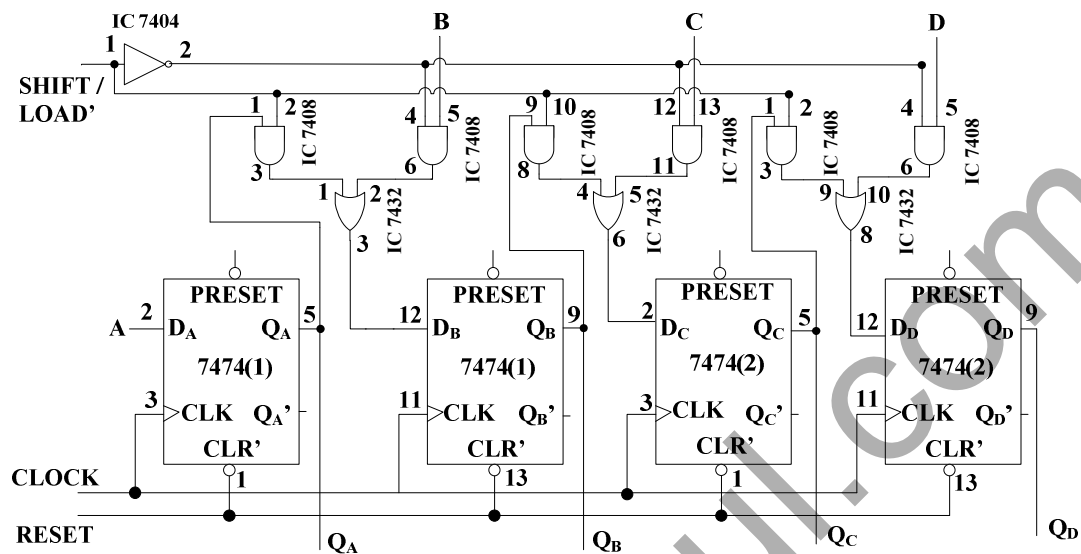
Truth Table for PIPO Shift Register :

Clk	Parallel input [A...D]	QA	QB	QC	QD	Parallel Output [QA...QD]
0	1011	0	0	0	0	0000
1	1100	1	0	1	1	1011
2	0000	1	1	0	0	1100

www.rejinpaul.com

Logic Diagram:

Parallel In Serial Out:



Truth Table for PISO Shift Register :

Shift/Load'	Clk	Parallel input [A...D]	Q _A	Q _B	Q _C	Q _D	Serial Output
0	1	1011	0	0	0	0	0
1	2	0000	1	0	1	1	1
1	3	0000	0	1	0	1	1
1	4	0000	0	0	1	0	0
1	5	0000	0	0	0	1	1
1	6	0000	0	0	0	0	0

Result:

Thus PISO and PIPO shift registers are implemented using flip flop with its truth table verified.

www.rejinpaul.com