State the working of a NOT gate circuit , which is implemented by using one BJT and resistances .

A NOT gate, also known as an inverter, produces an output that is the opposite (complementary) of its input. Using a Bipolar Junction Transistor (BJT) in a common-emitter configuration with resistors, you can create a simple NOT gate.

Here's a basic setup:

Components Needed:

BJT (NPN Type): Such as a transistor like 2N2222 or BC547.

Resistors: Two resistors, typically, one connected to the base and one connected to the collector of the transistor.

Circuit Design:

Base (B) of Transistor: Connect the input signal to the base of the NPN transistor through a resistor.

Collector (C) of Transistor: Connect a resistor from the collector to the power supply (VCC).

Emitter (E) of Transistor: Connect the emitter directly to the ground (GND).

Working Principle:

When the input voltage is low (logic 0), no current flows through the base of the transistor.

As a result, the transistor remains in a cutoff region, and no current flows through the collector.

This situation is equivalent to an open switch, causing the output voltage to be high (logic 1).

When the input voltage is high (logic 1), current flows through the base of the transistor.

The transistor switches into saturation, allowing current to flow from collector to emitter.

This effectively creates a low output voltage (logic 0).

Truth Table:

Input (A)	**Output (Q)**
0	1
1	0

This setup acts as a basic NOT gate, inverting the input signal. When the input is low, the output is high, and vice versa. A NOT gate, also known as an inverter, produces an output that is the opposite (complementary) of its input. Using a Bipolar Junction Transistor (BJT) in a common-emitter configuration with resistors, you can create a simple NOT gate.

Here's a basic setup:

Components Needed:

BJT (NPN Type): Such as a transistor like 2N2222 or BC547.

Resistors: Two resistors, typically, one connected to the base and one connected to the collector of the transistor.

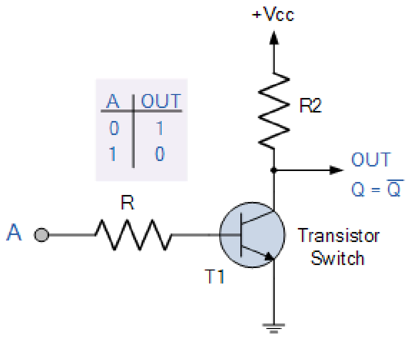
Circuit Design:

Base (B) of Transistor: Connect the input signal to the base of the NPN transistor through a resistor.

Collector (C) of Transistor: Connect a resistor from the collector to the power supply (VCC).

Emitter (E) of Transistor: Connect the emitter directly to the ground (GND).

Working Principle:



When the input voltage is low (logic 0), no current flows through the base of the transistor.

As a result, the transistor remains in a cutoff region, and no current flows through the collector.

This situation is equivalent to an open switch, causing the output voltage to be high (logic 1).

When the input voltage is high (logic 1), current flows through the base of the transistor.

The transistor switches into saturation, allowing current to flow from collector to emitter.

This effectively creates a low output voltage (logic 0).

Truth Table:

Input (A)	Output (Q)
0	1
1	0

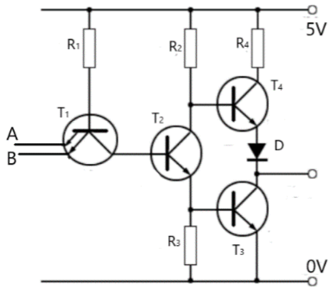
This setup acts as a basic NOT gate, inverting the input signal. When the input is low, the output is high, and vice versa.

Remember, the specific resistor values and transistor characteristics can affect the performance of the circuit. Calculations based on the transistor's datasheet might be necessary for precise design in practical applications.

Remember, the specific resistor values and transistor characteristics can affect the performance of the circuit. Calculations based on the transistor's datasheet might be necessary for precise design in practical applications.

Draw the TTL circuit diagrams NAND and NOR gate .

1. TTL NAND Gate Circuit Diagram:



The TTL NAND gate is typically constructed using multiple transistors arranged in a specific configuration to achieve the desired logic function.

Explanation:

The inputs (A and B) are connected to the bases of the transistors through resistors.

Transistors are connected in parallel, and their emitters are connected to the ground.

The collector outputs of these transistors are tied together and connected to the output (Q) through a resistor.

The arrangement of transistors and the resistor network allows the output to exhibit the logic behavior of a NAND gate.

2. TTL NOR Gate Circuit Diagram:

Similar to the NAND gate, a TTL NOR gate is constructed using multiple transistors arranged in a specific configuration to implement the NOR logic function.

Explanation:

The inputs (A and B) are connected to the bases of the transistors through resistors.

Transistors are connected in parallel, and their collectors are connected together.

The emitters of these transistors are tied to the VCC.

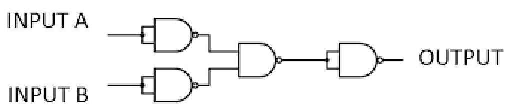
The collector output is connected to the output (Q) through a resistor.

This arrangement of transistors and the resistor network allows the output to exhibit the logic behavior of a NOR gate.

These circuit diagrams depict the basic implementation of TTL NAND and NOR gates using transistors. Actual TTL gates may involve additional components for stabilization, biasing, and to meet specific voltage and current requirements.

Design a 2 - input NOR gate circuit using only NAND gates .

- The circuit uses three NAND gates.
- Inputs A and B are connected to the first two NAND gates, each with their inputs inverted by connecting them to the NOT gates (additional inverters).
- The outputs of the first two NAND gates are fed into the third NAND gate, which acts as the final NOR gate.
- The output of the third NAND gate represents the logical NOR operation of the input signals A and B.


NOR gate from NAND gates

Write the logic expression for a 3 - input XOR gate and also obtain the corresponding K map .

A 3-input XOR gate produces an output that is high (1) when an odd number of its inputs are high (1), and it produces a low output (0) when an even number of its inputs are high.

The logic expression for a 3-input XOR gate is:

$$Y=A\oplus B\oplus C$$

Where:

�Y is the output.

�A, �B, and �C are the three input signals.

Now, let's create the Karnaugh map (K-map) for the 3-input XOR gate to help simplify the expression.

Karnaugh Map for 3-input XOR gate:

A B C	00	01	11	10
0	0	1	0	1

A B C	00	01	11	10
1	1	0	1	0

The entries in the K-map represent the output values for each combination of input states (A, B, C). Now, let's group the ones in the K-map to simplify the expression.

Grouping the Ones in the Karnaugh Map:

Grouping 1s:

$$A'C'+AB'C+ABC'+AC$$

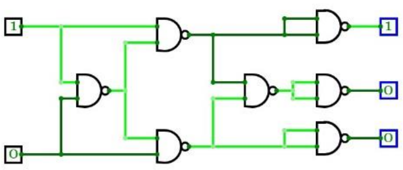
This expression represents the simplified form of the 3-input XOR gate logic.

Therefore, the logic expression for a 3-input XOR gate is:

$$Y=A'C'+AB'C+ABC'+AC$$

This expression denotes the Boolean logic function for a 3-input XOR gate based on the grouping of ones in its Karnaugh map.

Draw 1 - bit comparator circuit using only NAND gates



Explanation:

The circuit uses multiple NAND gates to compare two single-bit binary inputs, A and B.

It consists of three cascaded stages.

In the first stage:

NAND gates G1 and G2 compare the input bits A and B.

G1 produces the output A'B while G2 produces the output AB'.

In the second stage:

NAND gate G3 takes inputs A'B and AB' from the first stage. Its output is the LESS THAN (A<B) condition.

In the third stage:

NAND gate G4 takes inputs A and B directly.

NAND gate G5 takes inputs from G4 and the output of G3. Its output is the EQUALS (A=B) condition.

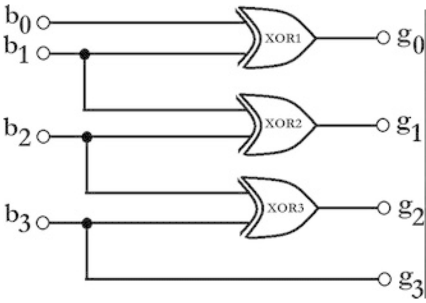
NAND gate G6 takes inputs A and B directly.

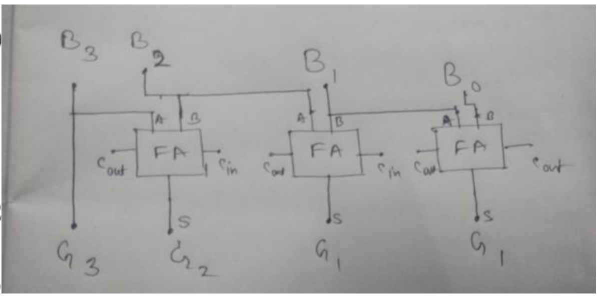
NAND gate G7 takes inputs from G6 and the output of G2. Its output is the GREATER THAN (A>B) condition.

The outputs of these NAND gates represent the comparison results: LESS THAN (A<B), EQUALS (A=B), and GREATER THAN (A>B).

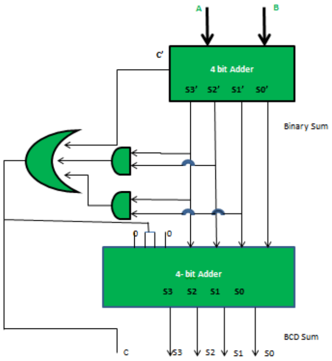
This circuit configuration implements the logic of a 1-bit comparator using only NAND gates. The comparison outputs indicate the relationship between the two 1-bit binary numbers A and B.

State how full adders can be used to convert one given binary number to corresponding gray code





State the working of a single digit BCD adder circuit with schematic circuit using 4 - bit adder IC and few basic gates



A Binary-Coded Decimal (BCD) adder is a digital circuit that performs addition specifically for BCD numbers. BCD numbers are decimal digits represented in binary form. A single-digit BCD adder typically involves adding two 4-bit BCD numbers (each digit represented by 4 bits) and generating a 4-bit sum along with a carry-out.

To create a single-digit BCD adder circuit, you can utilize a 4-bit adder IC (Integrated Circuit) and some basic logic gates like AND, OR, and XOR gates.

Working Principle:

Inputs:

Two 4-bit BCD numbers (each digit represented by 4 bits).

Carry-in (if necessary when chaining multiple adders).

Circuit Components:

4-bit Adder IC (such as the 74LS283 or equivalent).

Basic logic gates (AND, OR, XOR gates).

Connection wires and power supply.

Circuit Diagram (Schematic):

Circuit Explanation:

The two 4-bit BCD numbers (A and B) are given as inputs to the 4-bit adder IC.

The output of the 4-bit adder IC generates a 4-bit sum (S) and a carry-out (Cout).

Additional logic gates are used to handle BCD arithmetic and adjust the outputs to ensure proper BCD results.

Detailed Operation:

Addition Operation:

The 4-bit adder IC performs binary addition on inputs A and B.

It generates a 4-bit sum (S) and a carry-out (Cout).

BCD Adjustment:

The sum (S) output from the adder is checked for values greater than 9 (which are invalid in BCD).

If the sum is greater than 9 (1010 in binary), adjustments are made to bring the result back into valid BCD range.

Logic gates (AND, OR, XOR gates) are used to detect when the sum is greater than 9 in any nibble (group of 4 bits).

Adjustments are made by adding 6 (0110 in binary) to the invalid BCD value to produce a valid BCD value.

The carry-out (Cout) can be used to carry over to the next BCD digit if needed.

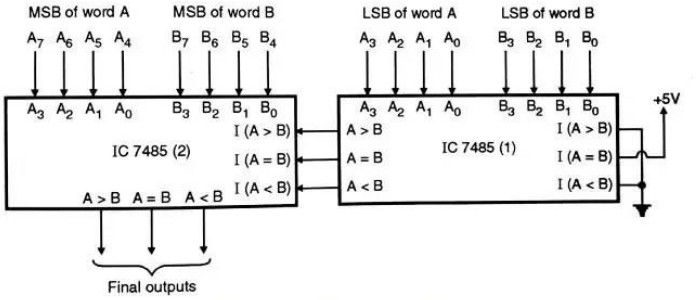
Output:

The corrected 4-bit sum represents the BCD result of the addition.

The carry-out (if generated) is used for chaining multiple BCD digits in multi-digit addition.

This circuit illustrates the process of adding two single-digit BCD numbers using a 4-bit adder IC and additional logic gates to handle BCD arithmetic, ensuring the outputs are valid BCD digits.

1) Use two 4 - bit comparator ICs to design a circuit in such a way that the same can be used for comparing two 8 - bit numbers. Show the schematic diagram with brief explanation of the circuit operation Details of comparator IC's and Pin diagram are not needed.



8-Bit Magnitude Comparator

Circuit Explanation:

Inputs:

Two 8-bit numbers (A and B) to be compared.

Components:

Two 4-bit comparator ICs.

Connection wires and power supply.

Circuit Operation:

The 8-bit number A and the 8-bit number B are input into two separate 4-bit comparator ICs, respectively.

The comparator ICs are designed to compare each corresponding pair of bits from A and B (from MSB to LSB).

The output of each comparator IC indicates the relationship between the bits being compared:

Less than (A < B)

Equal to (A = B)

Greater than (A > B)

By cascading two 4-bit comparator ICs, you compare the 8-bit numbers bit by bit. The outputs from the first 4-bit comparator IC serve as inputs to the second 4-bit comparator IC.

The output from the second comparator IC determines the overall relationship between the 8-bit numbers A and B:

If all individual bit comparisons are equal (A = B) for each corresponding pair of bits, the overall output will indicate equality.

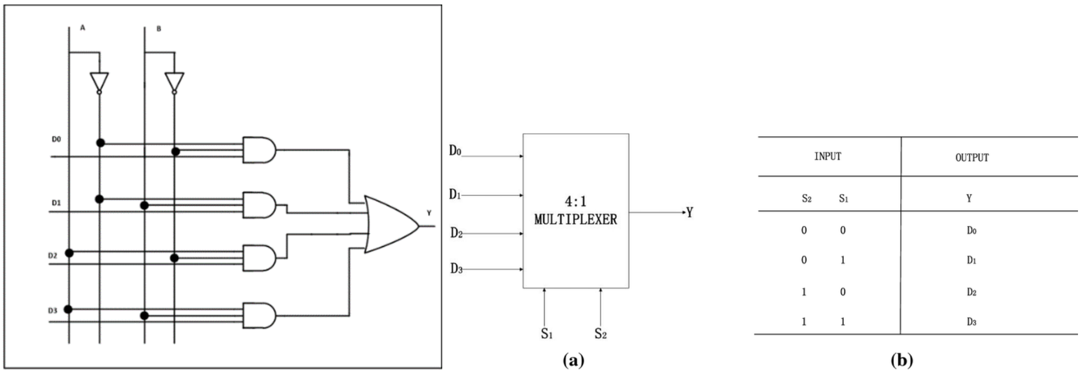
If any bit comparison shows A < B, the overall output will indicate A is less than B.

If any bit comparison shows A > B, the overall output will indicate A is greater than B.

The outputs of the individual bit comparisons and the overall comparison result provide the comparison relationship between the two 8-bit numbers.

This configuration using two 4-bit comparator ICs in a cascaded manner allows for the comparison of two 8-bit numbers, determining their relationship based on bit-to-bit comparisons, providing a comprehensive result regarding their relative magnitudes.

2) Design a 4:1 MUX using only basic gates. State its operation with the help of truth table.



INPUT		OUTPUT
S_2	S_1	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

4:1 MUX Circuit Diagram using Basic Gates:

Circuit Explanation:

This 4:1 MUX circuit is designed using basic gates (AND, OR, NOT).

It consists of:

4 input data lines (D0, D1, D2, D3)

2 select lines (S0, S1) used to choose among the data inputs.

1 output line (Y) that carries the selected data.

Operation with Truth Table:

Let's consider the truth table for a 4:1 MUX:

S1	S0	Output (Y)
0	0	D0
0	1	D1
1	0	D2
1	1	D3

Detailed Operation:

The select lines (S1, S0) determine which input data line gets connected to the output.

The truth table represents the behavior of the MUX based on the select lines.

For example:

When S1 = 0 and S0 = 0, the output Y will be the data present at input D0.

When S1 = 0 and S0 = 1, the output Y will be the data present at input D1.

When S1 = 1 and S0 = 0, the output Y will be the data present at input D2.

When S1 = 1 and S0 = 1, the output Y will be the data present at input D3.

Logic Implementation:

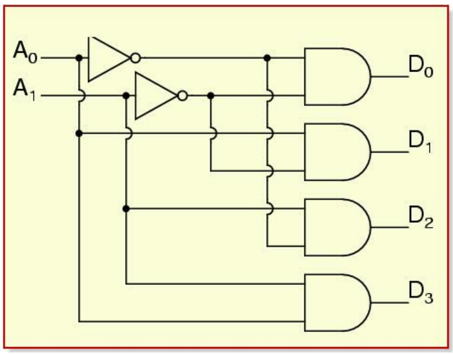
The circuit is designed using AND, OR, and NOT gates. The select lines S1 and S0 are used to generate the necessary combinations to select the desired input data line based on their binary values.

The gates are arranged to produce the logic needed to select the appropriate input data line based on the select inputs.

This 4:1 MUX circuit selects one of four input data lines and routes it to the output based on the binary values of the select inputs S1 and S0.

3) What is decoder? Design a 2-4 decoder circuit using basic gates .State how it can be converted into a 1:4 DeMUX.

A decoder is a combinational logic circuit used in digital electronics that takes an n-bit binary input and activates one of 2^n output lines based on that input. It essentially performs the opposite function of an encoder. Decoders are primarily used to decode a particular binary pattern and select a specific output line based on that pattern.



As a demultiplexer, a Decoder with Enable input might be used. A circuit known as a demultiplexer takes data from a single line and sends it to one of 2n different output lines.

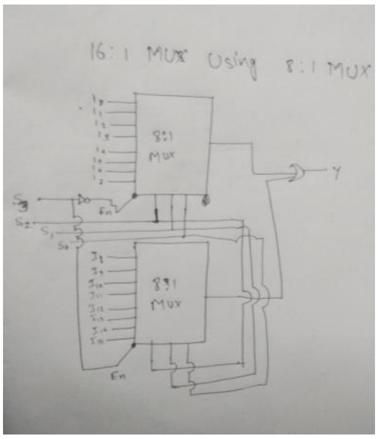
N selection lines and one input line are fed into a 2n demultiplexer. One is chosen using these selection lines instead of a potential 2n output line. We employ an n: 2n decoder with Enable input to create a 2n demultiplexer. The n selection lines of the Demultiplexer are the n input lines that the decoder receives, and the Demultiplexer’s single input line is the decoder enable input.

Using a 2:4 Decoder with Enable input to create a 1:4 demultiplexer. Let EN serve as the Demultiplexer’s input line and A and B as the selection lines.

When A and B are used as selection inputs, and EN is used as a data input line, the decoder depicted below performs as a 2:4 demultiplexer. Although the single input variable E has a path to all four outputs, the binary combination of the two selection lines A and B specifies that the input data is directed to only one of the output lines. The truth table of the circuit can be used to confirm this.

4) Draw the diode matrix encoder circuit for a numeric keyboard and explain its working briefly

5) IC 174151 is an 8:1 MUX chip complemented enable input (EN') and two output terminals (Y) and (Y') Cascade two such ICs to have one 16:1 MUX.


16:1 MUX Using 8:1 MUX
8:1 MUX
8:1 MUX
S3
S2
S1
S0
En
En
I10
I11
I12
I13
I14
I15
Y