Unit – 1

Session - 1

Digital Principles

Objectives

- Understand the difference between analog and digital signals
- Based on input conditions, determine the output of a buffer, a tri-state buffer, an inverter, a tri-state inverter, an AND gate, and an OR gate
- Recognize digital logic symbols

Introduction

Today, life is with electronics. Electronics has penetrated every aspect of everyone's life. Electronic circuits and systems are divided into two broad categories (based on the type of signals they process):

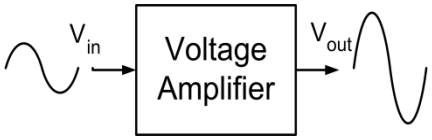
- Analog Electronic Circuits
- Digital Electronic Circuits

Logic design is the design of logic circuits which are basically digital electronic circuits. Electrical, electronics and computer engineers need to have a thorough understanding of logic circuits. These circuits are used to build digital computers, digital calculators, mobile phones, communication systems, and modern household appliances like Television, music systems, DVD players, etc.

Analog Electronic Circuits

Analog electronic circuits are designed for use with small signals. It exhibits linear operation.

Example: Voltage amplifier



The output voltage is the faithful amplified version of the input voltage signal.

Digital Electronic Circuits

Digital electronic circuits are used with large signals. It exhibits non-linear operation.

Example: Remote control circuit for automatic switching of a light

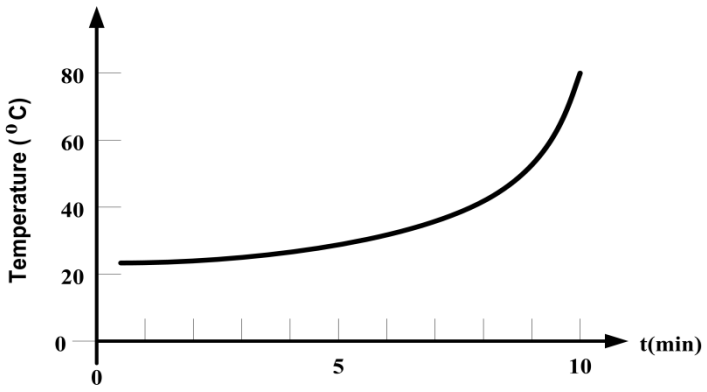
The signal is the current from a light-sensing circuit. The output signal is just ON or OFF and it is not an amplified version of the input signal.

Analog Signal

It is continuous and all possible values are represented (Continuous-time signal). Virtually all naturally occurring physical phenomena are analog signals. Temperature, Pressure, Velocity, and Speech (Sound) are signals that take on all possible values between given limits

Example:

Consider the temperature of water that is heated, which is continuously recorded. It changes smoothly from 23°C (room temp.) to 80°C as shown below:

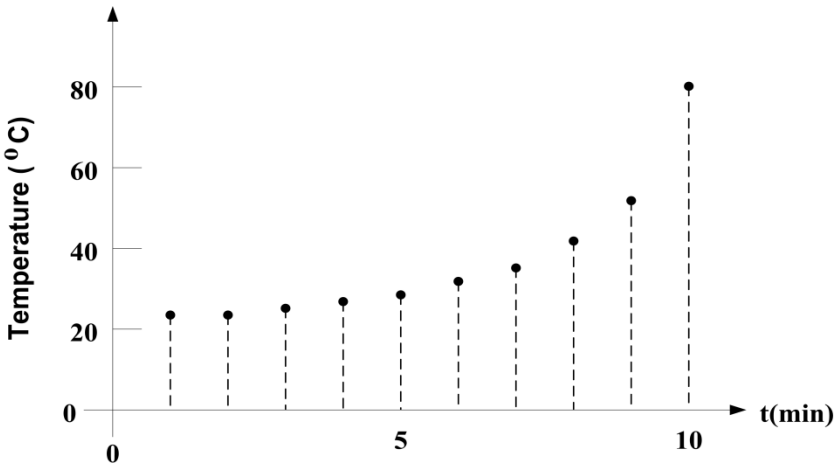


Discrete-Time Signal

Discrete-time signal is defined at discrete times. It may arise by sampling a continuous-time signal, or can be generated directly by some discrete-time process.

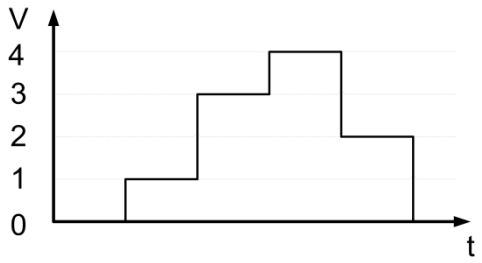
Example:

Consider that water temperature is measured and recorded only once every minute.

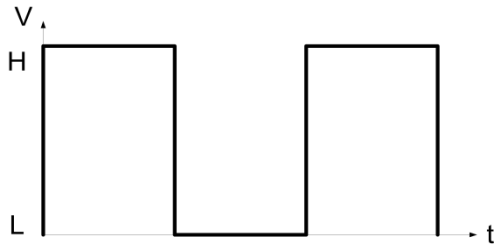


Digital Signal

In a digital signal both time and amplitude are discrete. It represents only a finite number of discrete values as shown in figure below:



A simple digital signal has only two discrete levels as shown in figure below:



The two discrete levels are represented as low level (L) and high level (H). It is called a logic (binary) signal.

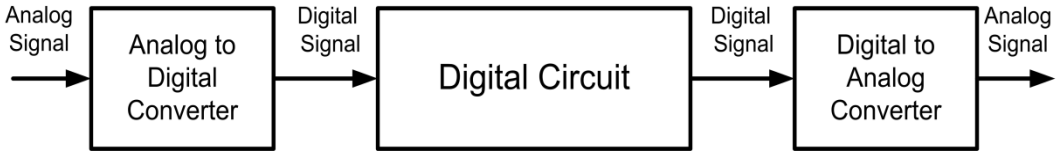
Digital Circuits and Systems

In digital circuits and systems, both the input and the output are digital signals. The digital circuits are widely used since they have the following advantages:

- The digital circuit elements usually operate in one of the two states – ON / OFF states, resulting in simple circuit operation.
- Digital circuits are realized as integrated circuits which are highly reliable, extremely small in size, and cost very less.
- Design of digital circuits needs knowledge of Boolean algebra and basic concepts of electrical network analysis that is simple and easy to learn.

But the signals around us are analog, for example, temperature, pressure, velocity, and speech signal. Processing analog signals is difficult. So we convert the analog signal into digital signal using an analog to digital converter (ADC) circuit. The digital signal is processed in a digital circuit. The resulting digital output signal is next converted back to analog signal using the digital-to-analog converter (DAC) circuit.

The complete process is as shown in the figure below:

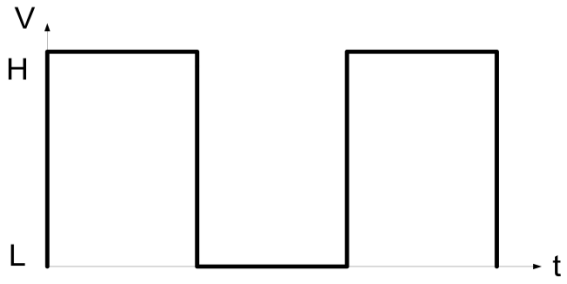


Binary System

Digital electronics today involve circuits that have exactly two possible states. A system having only two states is said to be binary (bi means two). The binary number system has exactly two values 0 and 1 and is widely used in digital electronics. The operation of a digital electronics circuit can be described in terms of its voltage levels. There are only two voltage levels. Clearly, one voltage is more positive than the other. The more positive voltage is the High (H) level, and other is the Low (L) level .

Ideal Digital Signal

The voltage levels in an ideal digital circuit will have values of either +5 V or 0 V. The voltages change (switch) between values, instantaneously. The ideal digital signal is as shown in figure below:

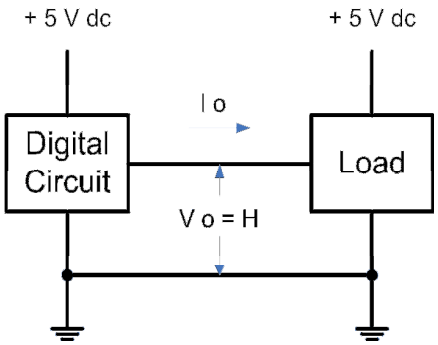


Digital Waveforms

The output voltage level of any digital circuit depends somewhat on its load. The circuit must act as a current source to deliver the current I_o to the load.

Loading of Digital Circuit (V_o= H)

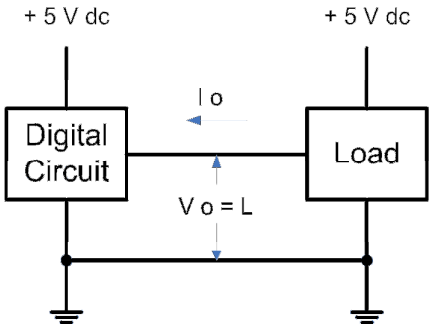
Consider the digital circuit connected to a load and the output voltage level is High (H) as shown in figure below:



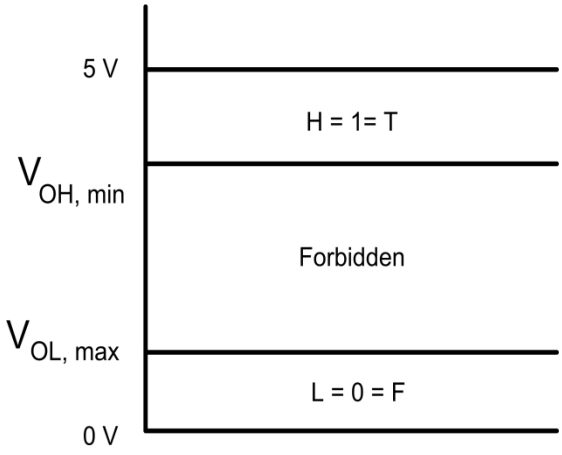
The circuit may not be capable of delivering the necessary I_o while maintaining +5 V. $V_{OH,min}$ is the minimum value of the output voltage when high.

Loading of Digital Circuit (V_o= L)

Consider the digital circuit connected to a load and the output voltage level is Low (L) as shown in figure below:

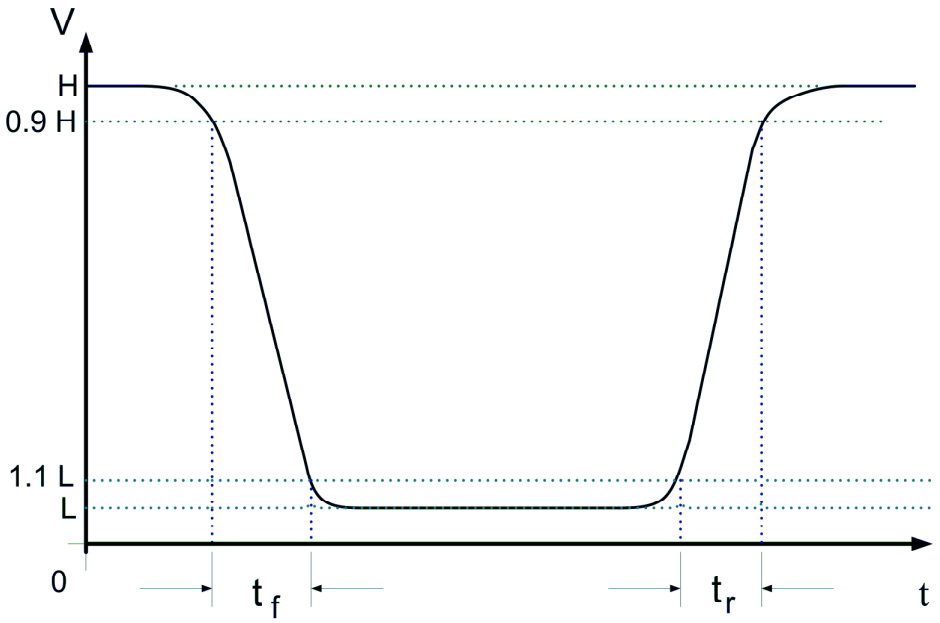


The digital circuit must act as a current sink. It must be capable of accepting a current I_o from the load and delivering to ground. $V_{OL,max}$ is the maximum value of the output voltage when low.



Switching Time

A finite amount of time is required for V_o to make the transition (switch) between levels. Consider the switching in a digital circuit as shown below:



Fall Time

The time required for V_o to make the transition from its high level to its low level is defined as fall time t_f. For ease of measurement we use 0.9 H and 1.1 L.

Example: If H = 4 V and L = 0.2 V

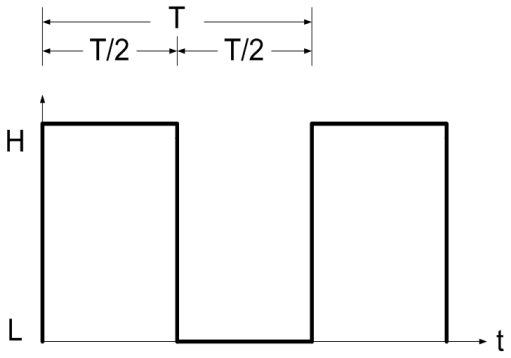
0.9 H = 3.6 V and 1.1 L = 0.22 V

Rise Time

The time required for V_o to make the transition from its low level to its high level is defined as rise time t_r It is measured between 1.1 L and 0.9 H.

Period and Frequency

There are many occasions where a symmetrical digital signal will be used. The electronic circuit used to generate this square wave is referred to as the **system clock.** Consider a symmetrical digital signal as shown:

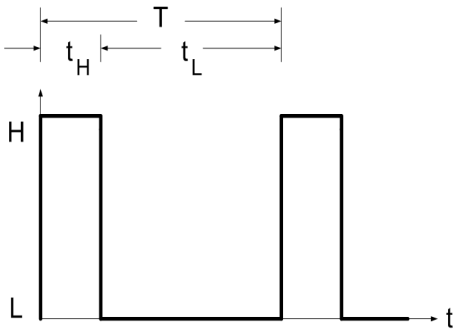


The period of waveform: **T**

The frequency **f = 1/T**

Duty Cycle

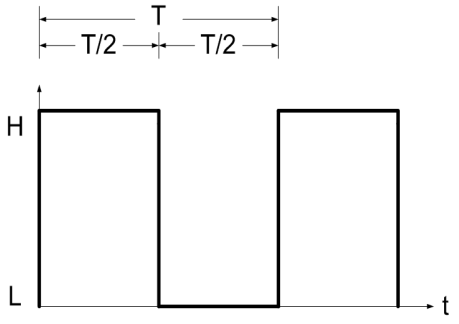
Duty cycle is a convenient measure of how symmetrical or how unsymmetrical a waveform is. Consider the waveform:



Duty cycle H = t_H / T

Duty cycle L = t_L/ T

Duty Cycle for Symmetrical Wave

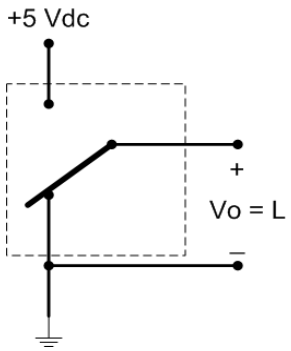


Duty cycle H = Duty cycle L = (T/2) / T = 0.5

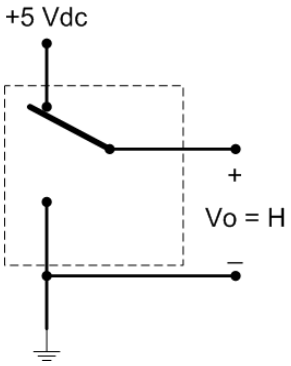
Digital Logic: Generating Logic Levels

The digital logic levels can be produced using switches as shown:

Switch is DOWN, V_o = L



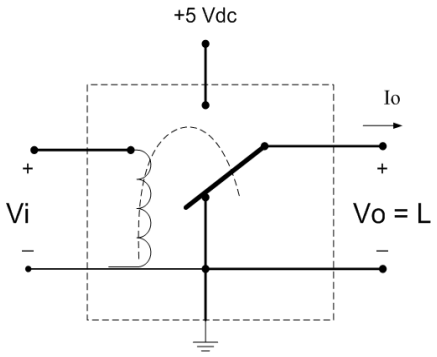
Switch is UP, V_o = H



Switch is easy to use and easy to understand, but it must be operated manually.

Generating Logic Levels using Relay

A relay is a switch that is actuated by applying a voltage V_i to a coil as shown:



The coil current develops a magnetic field that moves the switch arm from one contact to the other. Switches and relays were useful in the construction of early machines used for calculation and/or logic operations.

Disadvantages:

- Bulkier
- Cannot switch rapidly

They are replaced by digital ICs.

The Buffer

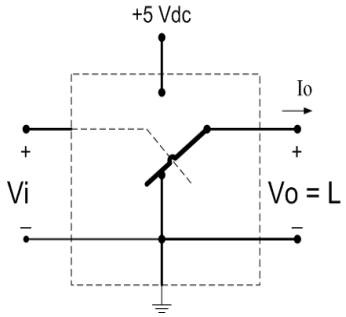
A buffer is an electronic switch. It is actuated by the input voltage V_i. Its operation is similar to the relay. The buffer is capable of delivering additional current to a load, hence the name buffer amplifier.

The Buffer Operation

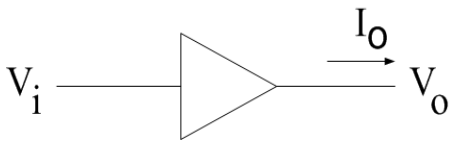
Truth Table:

V_i	V_o
0	0
1	1

Model:



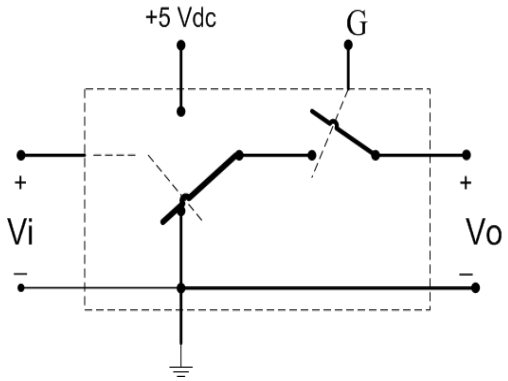
Symbol:



The Tri-state Buffer

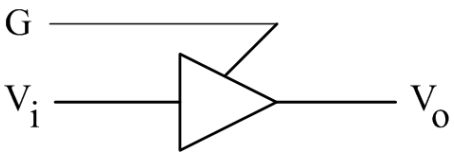
Consider a digital system with the possibilities: There may be more than one input signal and it is necessary to connect only one signal at a time. The output may need to be directed to more than one destination, one at a time. It is a simple buffer with an additional switch controlled by an input G.

Model:



When G is low (0), this switch is open and the output is disconnected from the buffer.

Symbol:



When G is high, the switch is closed and the output follows the input.

Truth Table:

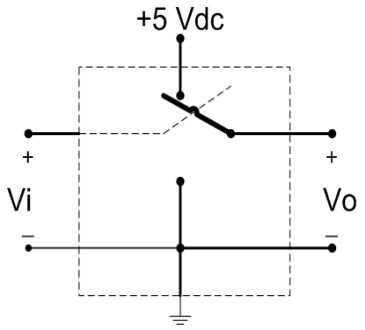
V_i	G	V_o
0	0	open
1	0	open
0	1	0
1	1	1

In effect, the control signal G connects the buffer to the load or disconnects the buffer from the load. Since it generates three types of signals, it is called *three-state buffer* or *tri-state buffer.*

The Inverter

One of the most basic operations in a digital system is inversion, or negation. This requires a circuit that will invert a digital signal. This logic circuit is called an inverter, or a NOT circuit.

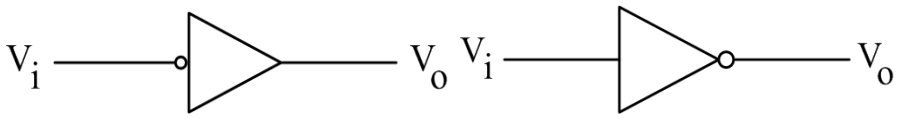
Model:



Truth table:

V_i	V_o
0	1
1	0

Symbols:



The Tri-State Inverter

When G is low, the inverter is connected to the output.

Truth Table:

V_i	G	V_o
0	0	1
1	0	0
0	1	open
1	1	open

The AND Gate

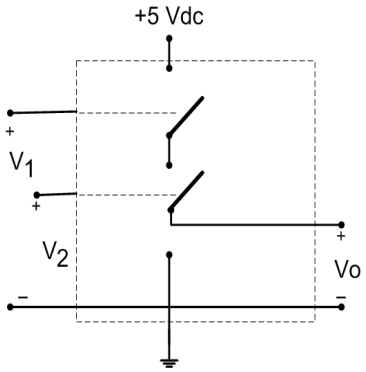
An AND gate is a digital circuit having two or more inputs and a single output. The AND gate has a high output only when all inputs are high.

Two- input AND Gate

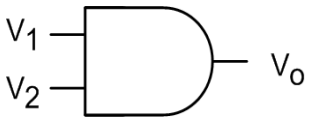
Truth table:

V_1	V_2	V_o
0	0	0
0	1	0
1	0	0
1	1	1

Model:



Symbol:



The OR Gate

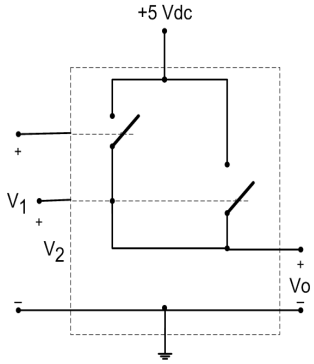
An OR gate is also a digital circuit having two or more inputs and a single output. The OR gate output is high if any or all of the input voltages are high.

Two-input OR gate

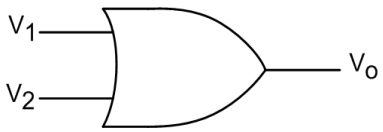
Truth table:

V_1	V_2	V_o
0	0	0
0	1	1
1	0	1
1	1	1

Model:



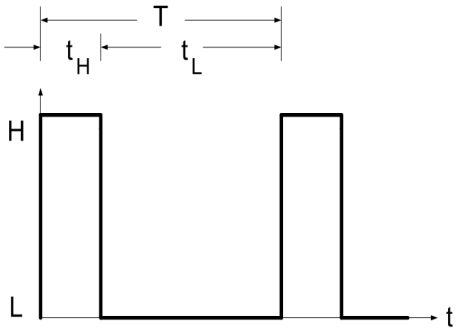
Symbol:



Questions

1. Give the characteristics of an analog signal and a digital signal.

2. The waveform given below has a duty cycle D = 0.2, and the positive pulses occur every 500 µs. What is the width of each positive pulse?



3. Construct a truth table for a 3-input AND gate.

4. What is a tri-state circuit?

Unit – 1

Session - 2

Digital Integrated Circuits

Objectives

- Review Digital Integrated Circuits
- Discuss IC Logic Families
- Understand Transistor-Transistor Logic (TTL)
- Describe TTL NAND gate working
- Be familiar with TTL Series and TTL Parameters

Introduction

A digital integrated circuit (IC) is constructed by an interconnection of resistors, transistors, and small capacitors all of which have been formed on the surface of a semiconductor wafer. The entire circuit resides on a tiny piece of semiconductor material called a chip.

The semiconductor wafer is typically a slice of monocrystalline silicon. The typical thickness is 0.2 mm and diameter is 8 to 15 cm. The wafer is divided into rectangular areas. Each area will become a single chip. The resistors and transistors necessary for each digital circuit are then formed on each chip by a series of semiconductor processing steps. After the processing steps are completed, the wafer is separated into individual chips. Each chip is then mounted in a suitable package.

IC Families

ICs are categorized by size according to the number of gates contained on each chip as shown:

Small-scale integration (SSI) IC	10 – 12 gates
Medium-scale integration (MSI) IC	12 – 100 gates
Large-scale integration (LSI) IC	100 -1000 gates
Very large-scale integration (VLSI) IC	> 1000 gates

Also, ICs are categorized according to the type of transistors used. The two basic transistor types are:

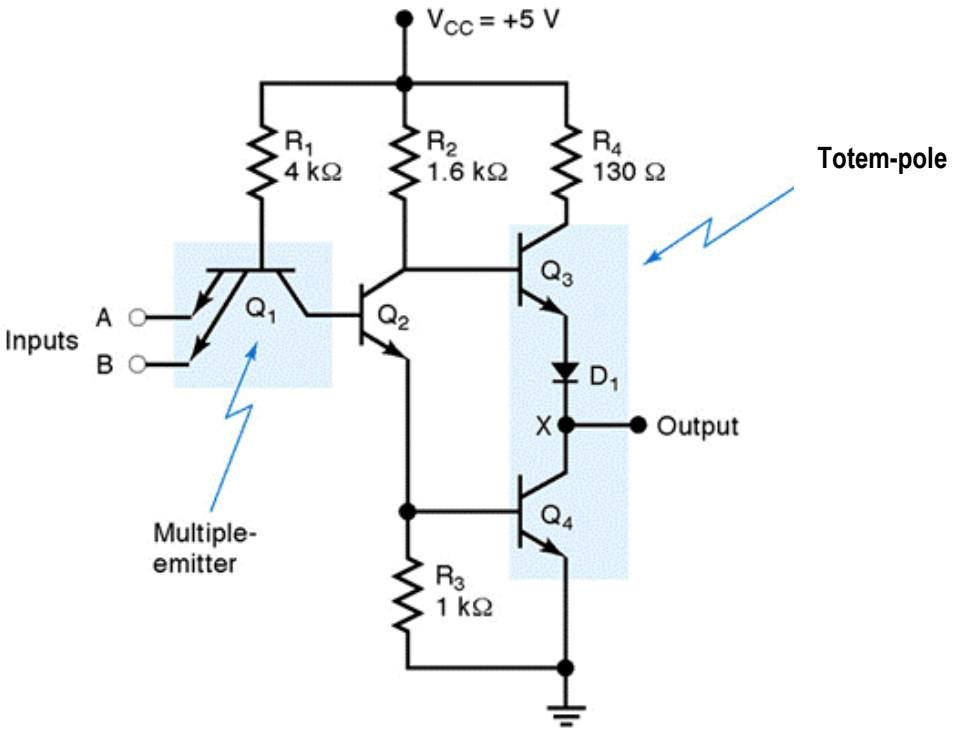
- Biploar
  - Bipolar Junction Transistor (BJT)
- Unipolar
  - Metal Oxide Semiconductor Field-Effect Transistor (MOSFET)

Transistor-Transistor Logic (TTL)

The transistor-transistor logic (TTL) is an important digital circuit family constructed using bipolar junction transistors. TTL was invented in 1961 by James L. Buie. Texas Instruments introduced the 5400 Series for military applications in 1964 and the 7400 Series for industrial applications later.

7400 TTL Series – Standard TTL

A standard TTL NAND gate (IC type no. 7400) is as shown below:



The circuit consists of multiple-emitter input transistor Q_1, phase-splitter transistor Q_2, and totem-pole transistors Q_3 and Q_4.

Circuit Operation

Inputs		Transistors State				Output
A	B	Q_1	Q_2	Q_3	Q_4	Y
0	0	On	Off	On	Off	1
0	1	On	Off	On	Off	1
1	0	On	Off	On	Off	1
1	1	Off	On	Off	On	0

It is clear that the circuit works as a two-input NAND gate.

Propagation Delay Time and Power Dissipation

The propagation delay time is the time it takes for the output of a gate to change after the inputs have changed. Propagation delay time is 10 nanoseconds and the power dissipation is 10 mW.

5400 Series

	5400 series	7400 series
Developed for	Military applications	Industry applications
Temperature range	-55 °C to 125 °C	0 °C to 70 °C
Supply voltage	4.5 V to 5.5 V	4.75 V to 5.25 V
Cost	High	Low

High-Speed TTL – 74Hxx series

By decreasing the resistances, propagation delay time is decreased. The smaller resistances, however, increases power dissipation. The propagation delay is 6 ns and power dissipation is 22 mW.

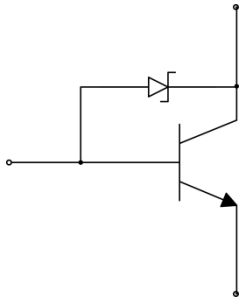
Low-Power TTL – 74Lxx Series

By increasing the internal resistances, the power dissipation of TTL gates can be reduced. The power dissipation is 1 mW and propagation delay time is 35 ns.

Schottky TTL – 74Sxx Series

In standard, high-speed, and low-power TTL the transistors are switched on with excessive current, causing a surplus of carriers to be stored in the base. To switch a transistor from on to off, we have to wait for the extra carriers to flow out of the base - the delay is known as saturation delay time. One way to reduce saturation delay time is with Schottky TTL.

Schottky diode *(named after German Physicist Walter H. Schottky)* connected as shown prevents transistor saturation. The power dissipation is 20 mW and the propagation delay time is 3 ns.



Low-Power Schottky TTL – 74LSxx Series

By increasing internal resistances as well as using Schottky diodes, we have a compromise between low power and high speed. Thus we obtain the low power Schottky TTL – a favorite of digital designers. The power dissipation is 2 mW and the propagation delay time is 10 ns.

Questions

1. With the aid of a circuit diagram, explain the operation of a 2 - input TTL NAND gate with totem - pole output.

2. Discuss the features of high speed TTL, low power TTL, Schottky TTL families.

Unit – 1

Session - 3

TTL Parameters

Objectives

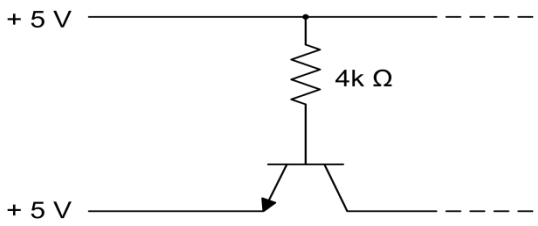
- Understanding various TTL Parameters
  - Floating Inputs
  - Worst-Case Input Voltages & Output Voltages
  - Profiles and Windows
  - Compatibility
  - Sourcing and Sinking
  - Noise Immunity
  - Standard Loading and Loading Rules

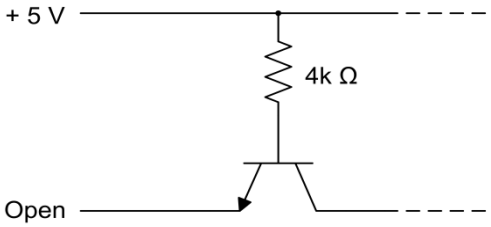
Introduction

The TTL series devices are guaranteed to work reliably over a temperature range of 0 to 70°C and over a supply voltage of 4.75 to 5.25 Volts. We need to know the worst case values for the different parameters like input current, output voltage, and so on.

Floating Inputs

If TTL input is floating (unconnected) there is no emitter current. If TTL input is connected to logic 1 (+ 5 V), obviously, there is no emitter current. Hence a floating TTL input is equivalent to a logic 1.



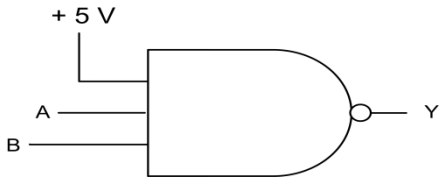


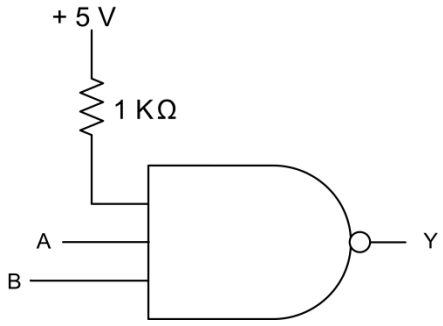
Disadvantage of Floating Inputs

- Floating input acts as a small antenna
- It will pick up stray electromagnetic noise voltages
- Noise pickup can be large enough to cause erratic operation of logic circuits

Solution

We connect unused TTL inputs to supply voltage. We have direct connection to supply voltage in case of 74S and 74LS series and high input through a pull-up resistor for 74, 74L, 74H series as shown below.

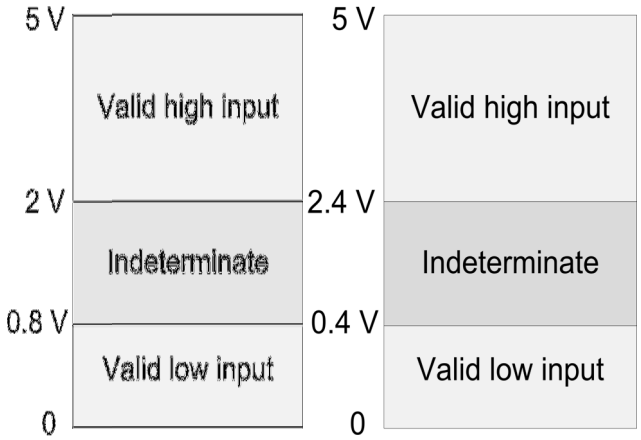




Worst-Case Input Voltages

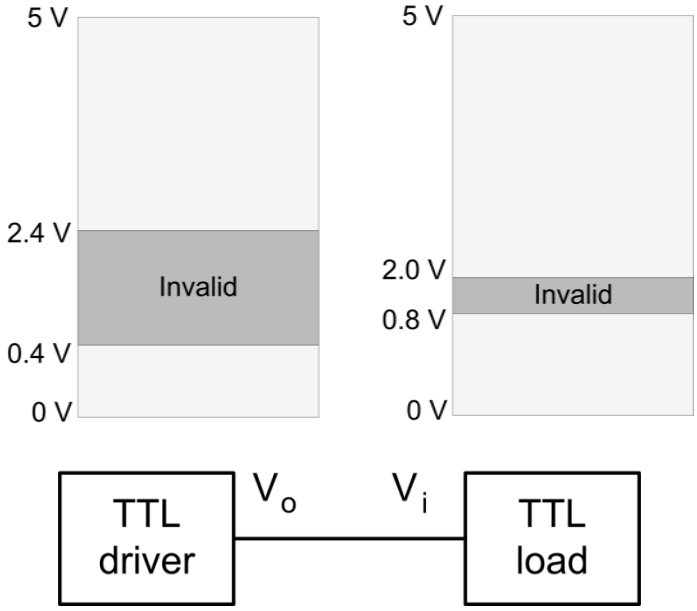
Consider a TTL inverter. Ideally, the low input state $\mathbf{V_{IL} = 0\ V}$ and the high input state $\mathbf{V_{IH} = 5\ V.}$ $\mathbf{V_{IL,\ max} = 0.8\ V}$ (*If the input voltage is greater than 0.8 V, the output state is unpredictable*). $\mathbf{V_{IH,\ min} = 2\ V}$ *(If the input voltage is less than 2 V, the output state is again unpredictable).* Ideally, the low output state $\mathbf{V_{OL} = 0\ V}$ and the high output state $\mathbf{V_{OH} = 5\ V.}$ $\mathbf{V_{OL,\ max} = 0.4\ V}$ and $\mathbf{V_{OH,\ min} = 2.4\ V}$.

TTL Input Profile



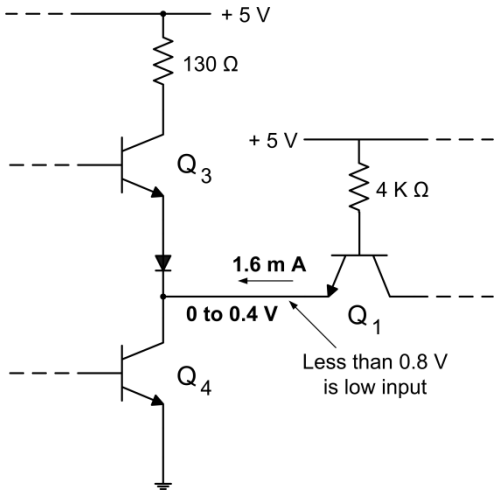
Compatibility

The TTL devices are compatible. The low and high output windows fit inside the low and high input windows, as shown in the figure below. Output of any TTL device is suitable for driving the input of another TTL device.



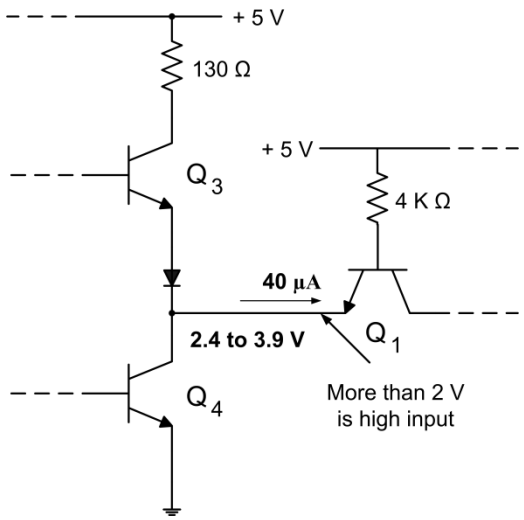
Sinking Current

When standard TTL output is low, transistor Q4 acts as a current sink and $I_{IL,\,max}$ = - 1.6 mA (see figure below).



Sourcing Current

When the standard TTL output is high a reverse emitter current flows, transistor Q3 acts as a source and $I_{IH, max}$ = 40 µA (see figure below).

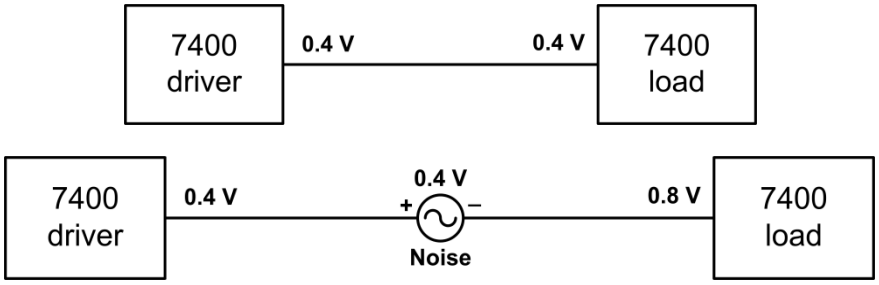


Noise Immunity

The worst-case low values are:

$V_{OL, max}$ = 0.4 V driver output

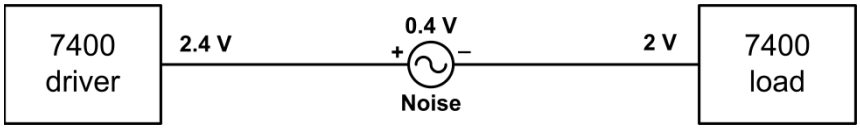
$V_{IL, max}$ = 0.8 V load input



The worst-case high values are:

$V_{OH, min}$ = 2.4 V driver output

$V_{IH, min}$ = 2 V load input



There is a difference of 0.4 V between the driver output voltages and required load input voltages.

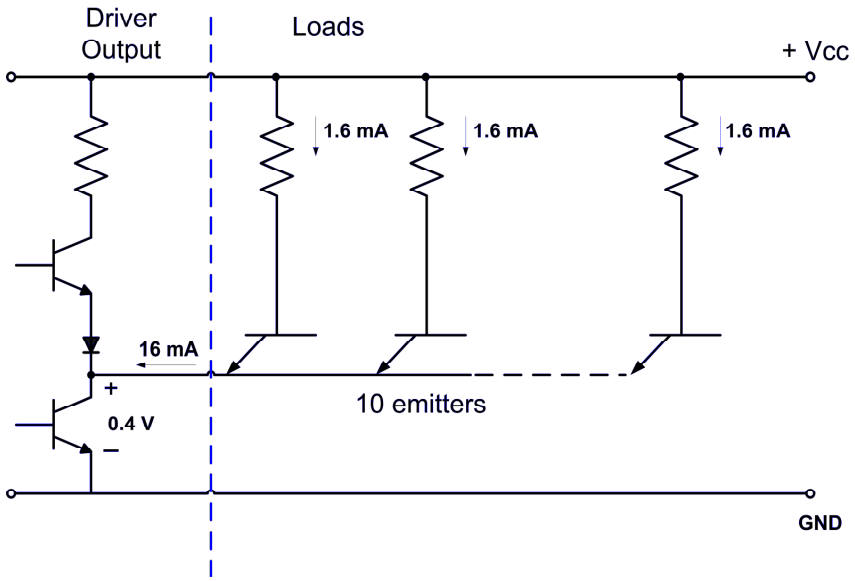
This difference is called noise immunity. The noise immunity values Δ0 and Δ1 corresponding to low level and high level are calculated as shown below:

Noise Immunity Δ0	$V_{IL, max} - V_{OL, max}$	0.8 – 0.4 = 0.4 V
Noise Immunity Δ1	$V_{OH, min} - V_{IH, min}$	2.4 – 2 = 0.4 V

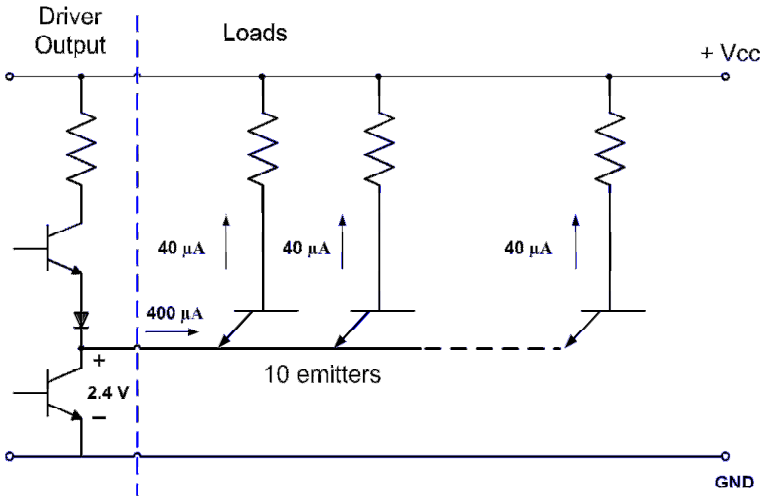
Standard Loading

A TTL device can source current (high output) or sink current (low output). 7400 series device can sink up to 16 mA i. e. $I_{OL, max}$ **= 16 mA.** It can source up to 400 µA i. e. $I_{OH, max}$ **= - 400 µA.** The worst-case TTL input currents are: $I_{IL, max}$ **= - 1.6 mA** and $I_{IH, max}$ **= 40 µA.** Output currents = **10 x** Input currents. So we can connect up to 10 TTL inputs to any TTL output and the fanout = 10.

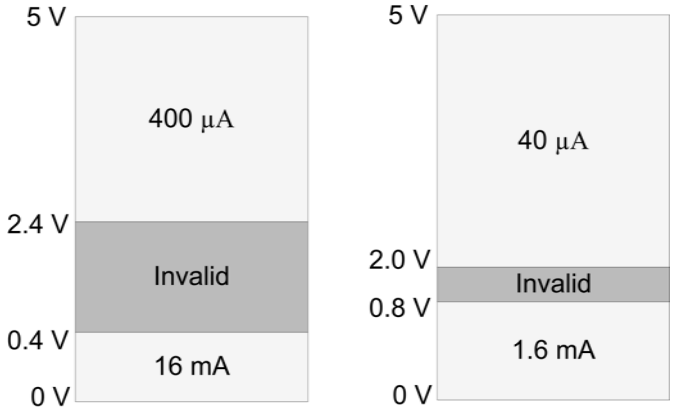
Low-state Fanout



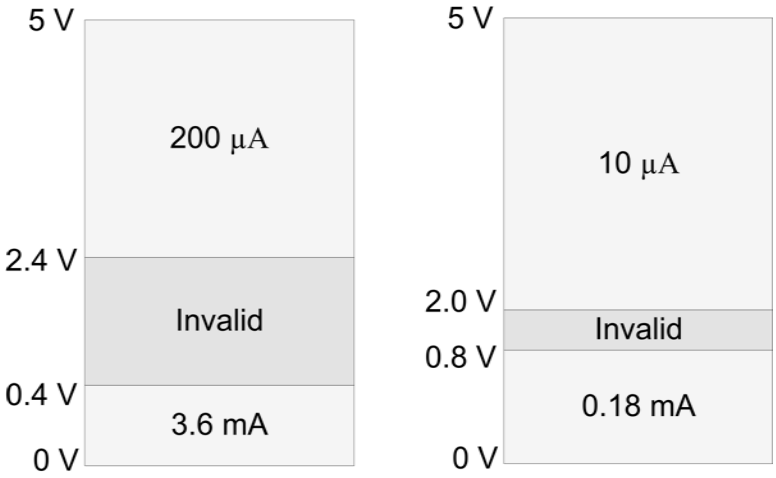
High-state Fanout



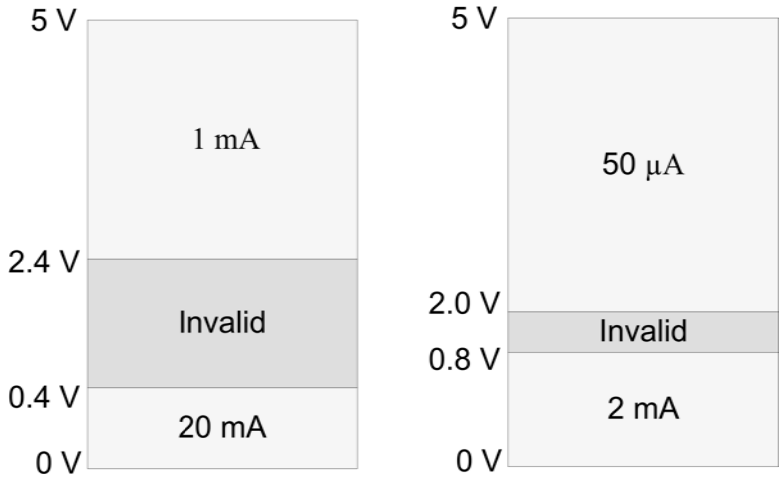
Standard TTL Output-Input Profile



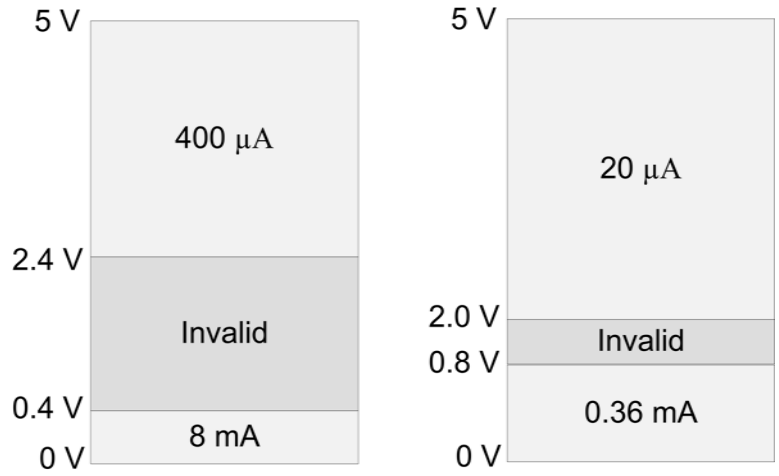
Low-power TTL Output-Input Profile



Schottky TTL Output-Input Profile



Low-Power Schottky TTL Output-Input Profile



Fanouts of TTL Types

TTL Driver	TTL Load				
	74	74H	74L	74S	74LS
74	10	8	40	8	20
74H	12	10	50	10	25
74L	2	1	20	1	10
74S	12	10	100	10	50
74LS	5	4	40	4	20

Questions

1. Discuss the importance of the various TTL parameters.

2. What is the fanout of a 74S00 device when it drives low-power TTL load?

Unit – 1

Session - 4

Digital Logic

Objectives

- Understand what are logic gates
- Review Boolean Algebra
- Write the truth tables for, and draw the symbols for, 2-input OR, AND, NOR, and NAND gates
- Write Boolean equations for logic circuits and draw logic circuits for Boolean equations
- Use De Morgan's first and second theorems to create equivalent circuits
- Positive and Negative Logic

Introduction

A digital electronic circuit having one or more input signals, but only one output signal is called a gate. Gates simulate mental processes; hence they are called logic gates. A logic gate performs a logical operation on one or more logic inputs and produces a single logic output.

Review of Boolean Algebra

Boolean algebra was developed in 1854 by George Boole. In 1937 Claude Shannon showed that Boolean algebra can be used to describe logic circuits. Boolean algebra is widely used in the design of digital circuits and computers. It is the mathematical foundation of logic design.

Boolean Algebra Definition

A Boolean algebra is an algebra consisting of a set B (which contains at least two elements 0 and 1) together with three operations : AND, OR, NOT defined on the set, such that for any element x and y of B, **x . y** (the product of **x** and **y**), **x + y** (the sum of **x** and **y**) and **x'** (the complement of **x**) are in B.

Axioms of Boolean Algebra

0 . 0 = 0	1 + 1 = 1
1 . 1 = 1	0 + 0 = 0
0 . 1 = 0	1 + 0 = 1
1 . 0 = 0	0 + 1 = 1

If x = 0, then x' = 1 and If x = 1, then x' = 0.

Single-Variable Theorems

If x is a Boolean variable in set B, then we have:

x . 0 = 0	x + 1 = 1
x . 1 = x	x + 0 = x
x . x = x	x + x = x
x . x' = 0	x + x' = 1
x'' = x (Double Inversion Rule)	

Note the principle of duality

Two - and Three – Variable Properties

x . y = y . x	Commutative
x + y = y + x	
x . (y . z) = (x . y) . z	Associative
x + (y + z) = (x + y) + z	
x . (y + z) = x . y + x . z	Distributive
x + (y . z) = (x + y) . (x + z)	
x + x . y = x	Absorption
x . (x + y) = x	
x . y + x . y' = x	Adjacency Theorem
(x + y) . (x + y') = x	
(x . y)' = x' + y'	De Morgan's Theorem
(x + y)' = x' . y'	
x + x' . y = x + y	Elimination
x . (x' + y) = x . y	
x . y + x' . z + y. z = x . y + x' . z	Consensus Theorem
(x + y) . (x' + z) . **(y + z)** = (x + y) . (x' + z)	

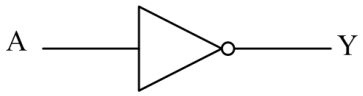
The Basic Gates – NOT, OR, AND

The basic logic gates are:

- The NOT gate (inverter)
- The OR gate
- The AND gate

The Inverter (NOT Gate)

Logic Symbol:

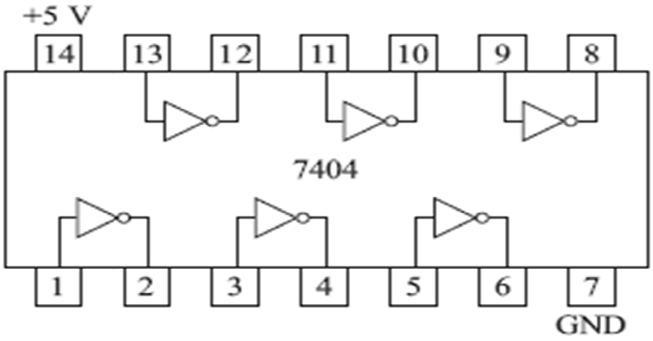


Truth Table:

Input	Output
A	Y
0	1
1	0

7404 TTL Hex Inverter

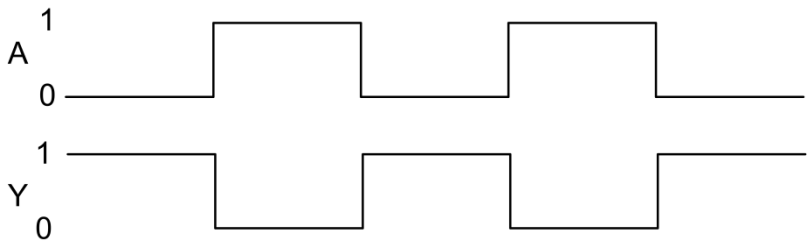
- The pin out diagram of a 7404 hex inverter is as shown:



It has six inverters (NOT gates).

Timing Diagram

Timing diagram is a picture of the input and output waveforms of a digital circuit. The timing diagram for inverter is as shown:

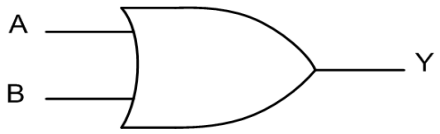


OR Gates

An OR gate has two or more input signals but only one output signal. It is called an OR gate because the output voltage is high if any **or** all the input voltages are high.

2-input OR Gate

Logic Symbol:



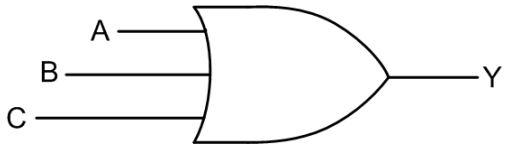
Y = A + B

Truth table:

Inputs		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

3-input OR Gate

Logic Symbol:



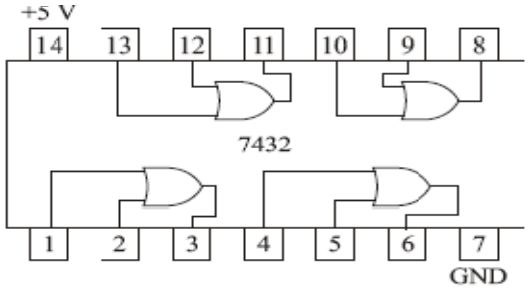
Y = A + B + C

Truth table:

Inputs			Output
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

7432 Quad 2-input TTL OR Gate

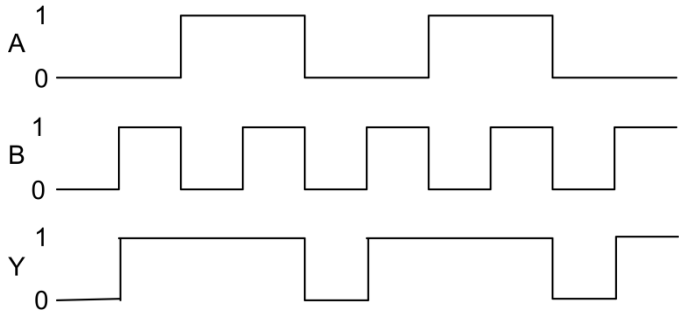
The pin out diagram of a 7432, a TTL quad 2-input OR gate is as shown:



It has four 2 input OR gates.

Timing Diagram of 2-input OR Gate

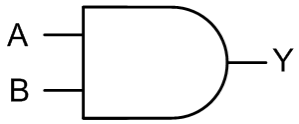
The timing diagram for a 2-input OR gate, when the two inputs given are as shown:



AND Gates

The AND gate has a high output only when all inputs are high.

Logic Symbol:



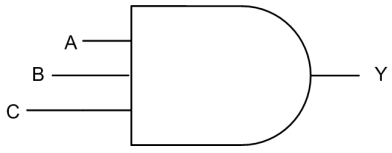
Y = A . B

Truth table of 2-input AND gate:

Inputs		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

3-input AND Gate

Logic Symbol:



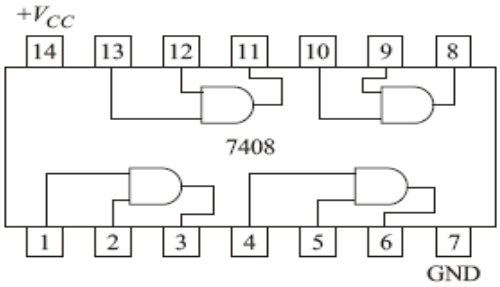
Y = A . B . C

Truth table:

Inputs			Output
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

7408 Quad 2-input AND Gate

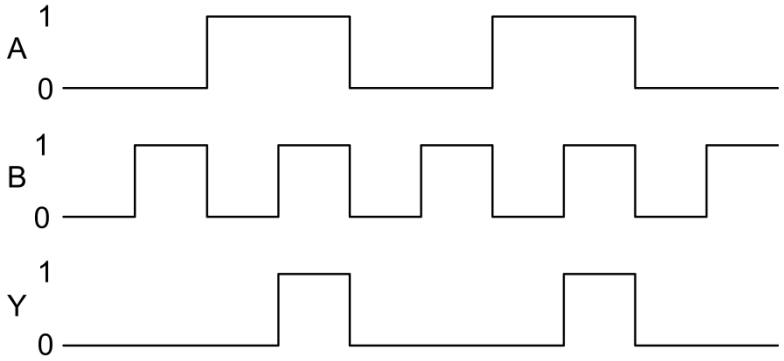
The pin out diagram of a 7408, a TTL quad 2-input AND gate is as shown:



It has four 2 input AND gates.

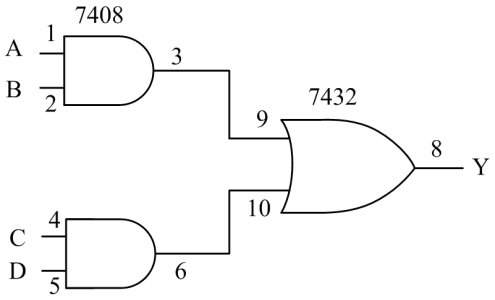
Timing Diagram of 2-input AND Gate

The timing diagram for a 2-input AND gate, when the two inputs given are as shown:



AND-OR Network

Consider the logic circuit as shown:



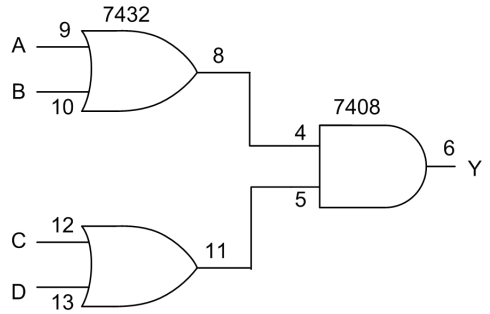
Y = A . B + C. D

This logic circuit is called an AND-OR network because input AND gates drive an output OR gate.

AND-OR networks always produce sum-of-products (SOP) equations.

OR-AND Network

Consider the logic circuit as shown:



Y = (A + B) . (C + D)

This logic circuit is called an OR-AND network because input OR gates drive an output AND gate.
OR-AND networks always produce product-of-sums (POS) equations.

Universal Logic Gates – NOR, NAND

Basic logic gates (NOT, AND, OR) can be used to realize any logic expressions. Is it possible to use only one type of gate for this purpose? Fabrication of IC that performs a logic operation becomes easier when gate of only one kind is used. A logic gate is called a universal logic gate, when any logic function can be realized using this one gate type. In fact, all logic functions can be constructed with only a single gate type, if that gate is inverting.

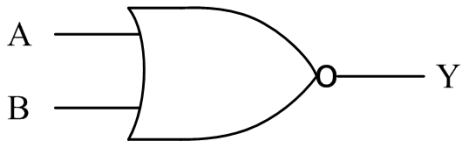
The two common inverting gates:

- NOR (inverted OR gate)
- NAND (inverted AND)

NOR and NAND gates are universal logic gates.

2-input NOR Gate

Logic Symbol:



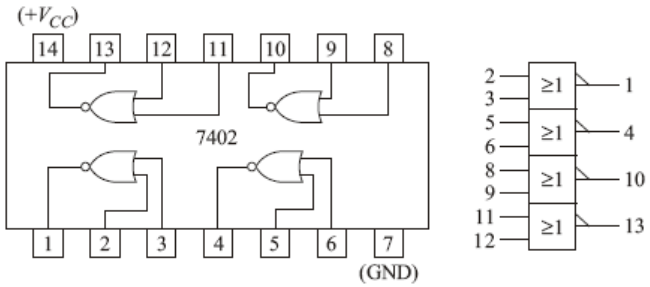
Y = (A + B)'

Truth Table:

Inputs		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

7402 Quad 2-input NOR Gate

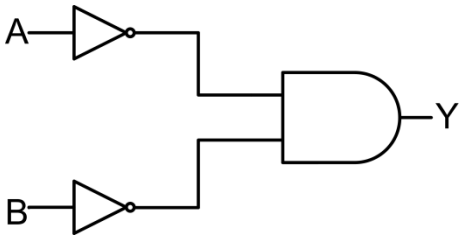
The pin out diagram of a 7402, a TTL quad 2-input NOR gate is as shown:



It has four 2-input NOR gates.

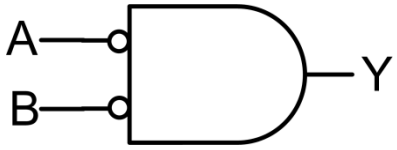
Bubbled AND Gate

Consider AND gate with inverted inputs and its equivalent symbol as shown:



Using De Morgan's First Theorem we have:

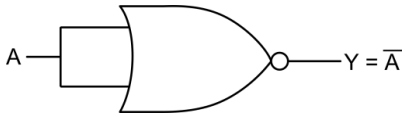
$$(A + B)' = A'. B'$$



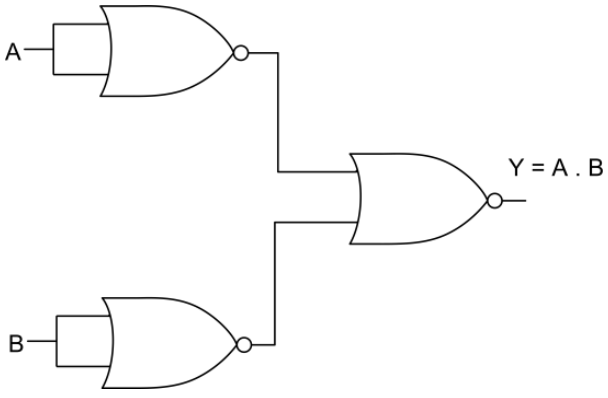
NOR gate is equivalent to a bubbled AND gate.

Universality of NOR Gates

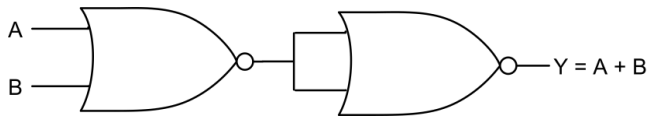
NOT operation using NOR gate is realized as shown:



AND operation using NOR gate is realized as shown:



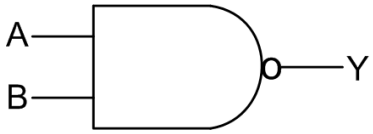
OR operation using NOR gate is realized as shown:



The basic logic operations – NOT, AND, OR, can be realized using NOR gate; hence NOR gate is called an universal logic gate.

2-input NAND Gate

Logic Symbol:



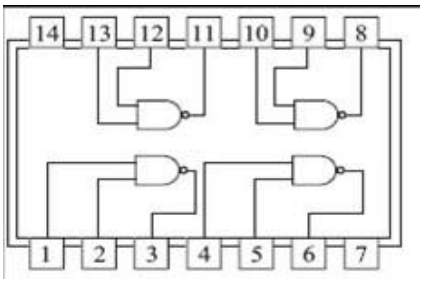
Y = (A . B)'

Truth Table:

Inputs		Output
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

7400 Quad 2-input NAND Gate

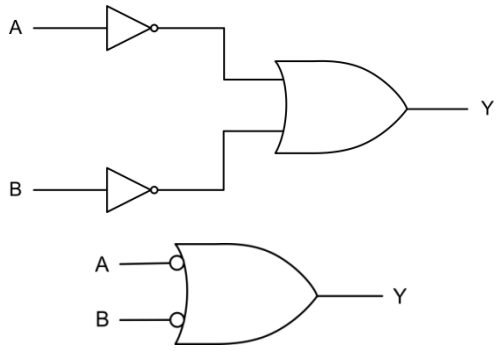
The pin out diagram of a 7400, a TTL quad 2-input NAND gate is as shown:



It has four 2-input NAND gates.

Bubbled OR Gate

Consider OR gate with inverted inputs and its equivalent symbol as shown:



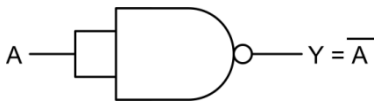
Using De Morgan's second theorem we have:

$$(A \cdot B)' = A' + B'$$

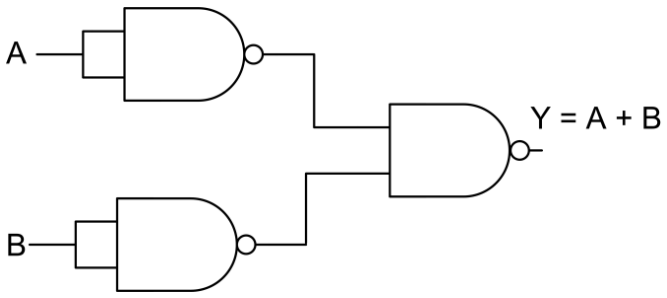
NAND gate is equivalent to a bubbled OR gate.

Universality of NAND Gate

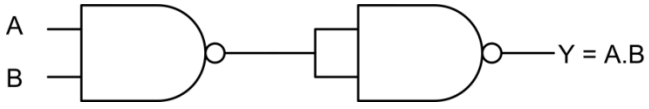
NOT operation using NAND gate is realized as shown:



OR operation using NAND gate is realized as shown:



AND operation using NAND gate is realized as shown:



The basic logic operations – NOT, AND, OR, can be realized using NAND gate; hence NAND gate is called an universal logic gate.

Positive and Negative Logic

In positive logic

- Low voltage (ex. 0 V) – Logic 0
- High voltage (ex. + 5 V) – Logic 1

In negative logic

- High voltage (ex. 0 V) – Logic 0
- Low voltage (ex. - 5 V) – Logic 1

Note that the device should turn ON for logic 1 input and turn OFF for logic 0 input.

Truth table of OR gate:

Inputs		Output
A	B	Y
Low	Low	Low
Low	High	High
High	Low	High
High	High	High

Truth table of OR gate (positive logic):

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Truth table of OR gate (negative logic):

A	B	Y
1	1	1
1	0	0
0	1	0
0	0	0

An OR gate in a positive logic system becomes an AND gate in a negative logic system.

Equivalences

Positive OR	↔	Negative AND
Positive AND	↔	Negative OR
Positive NOR	↔	Negative NAND
Positive NAND	↔	Negative NOR

Questions

1. What are universal gates? Realize basic gates using only NAND gates.

2. Implement the following function using universal gates only: (((A + B).C)').D

3. Explain the significance of DeMorgan's theorem.

4. Explain the principle of duality

5. Implement AB + C'D' with only three NAND gates. Draw logic diagram also. Assume inverted input is available.

6. Build the logic circuit whose output expression is Y = A . B . C + A . B . C

7. Build the logic circuit whose output expression is Y = (A + B + C) . (A + B + C)