AIM

Comparative study and making of Full 4-bit adder circuit.

CONTENTS

LOGIC GATES

Logic gate is a device that acts as a building block for digital circuits. They perform basic logical functions that are fundamental to digital circuits. Most electronic devices we use today will have some form of logic gates in them. For example, logic gates can be used in technologies such as smartphones, tablets or within memory devices.

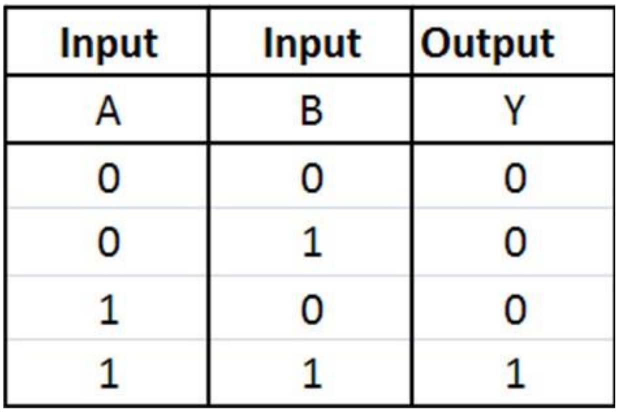
In a circuit, logic gates will make decisions based on a combination of digital signals coming from its inputs. Most logic gates have two inputs and one output. Logic gates are based on Boolean algebra. At any given moment, every terminal is in one of the two binary conditions, *false* or *true*. False represents 0, and true represents 1. Depending on the type of logic gate being used and the combination of inputs, the binary output will differ. A logic gate can be thought of like a light switch, wherein one position the output is off -- 0, and in another, it is on -- 1. Logic gates are commonly used in integrated circuits (IC).

BASIC LOGIC GATES

1) AND GATE
2) OR GATE
3) XOR GATE
4) NOT GATE
5) NAND GATE
6) NOR GATE
7) XNOR GATE

AND GATE

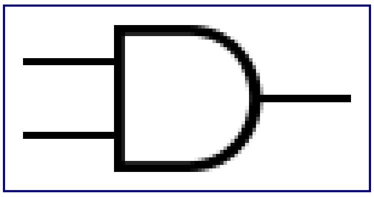
The AND gate is a basic digital logic gate that implements logical conjunction - it behaves according to the truth table below. A HIGH output Results only if both the inputs to the AND gate are HIGH. If neither or only one input to the AND gate is HIGH, a LOW output results. In another sense, the function of AND effectively finds the minimum between two binary digits. Therefore, the output is always 0 except when all the inputs are 1.



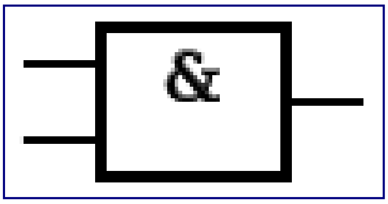
Input	Input	Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

SYMBOLS

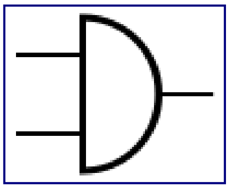
There are three symbols for AND gates: the American (ANSI or 'military') symbol and the IEC ('European' or 'rectangular') symbol, as well as the deprecated DIN symbol.



MIL/ANSI Symbol



IEC Symbol

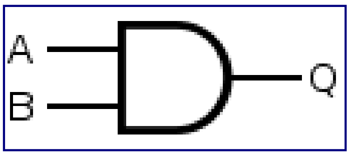


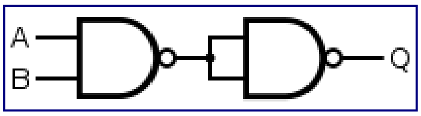
DIN Symbol

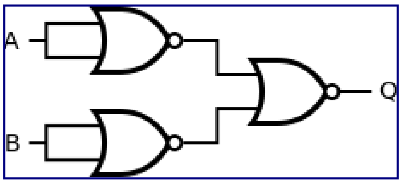
The AND gate with inputs *A* and *B* and output *C* implements the logical expression **C = A.**

ALTERNATIVES

If no specific AND gates are available, one can be made from NAND or NOR gates, because NAND and NOR gates are considered the "universal gates,"meaning that they can be used to make all the others.

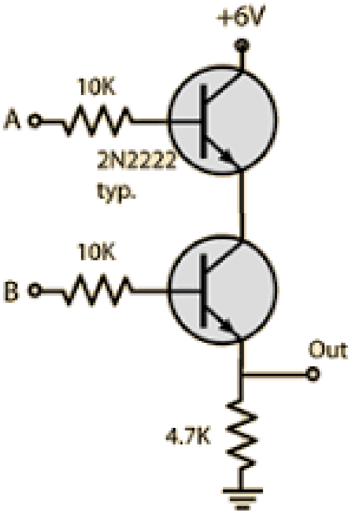






Desired gate **NAND construction** **NOR construction**

(CIRCUIT DIAGRAM OF AND GATE USING TRANSISTORS)



IC FOR AND GATE: 74(LS/HC)08

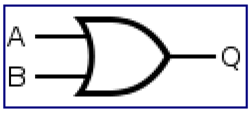
OR GATE

The OR gate is a digital logic gate that implements logical disjunction - it behaves according to the truth table below. A HIGH output (1) results if one or both the inputs to the gate are HIGH (1). If neither input is high, a LOW output (0) results. In another sense, the function of OR effectively finds the maximum between two binary digits, just as the complementary AND function finds the minimum.

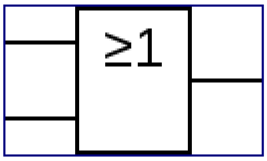
INPUT		**OUTPUT**
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

SYMBOLS

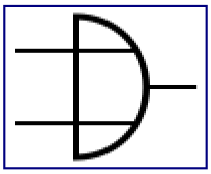
There are two symbols of OR gates: the American (ANSI or 'military') symbol and the IEC ('European' or 'rectangular') symbol, as well as the deprecated DIN symbol.



MIL/ANSI Symbol

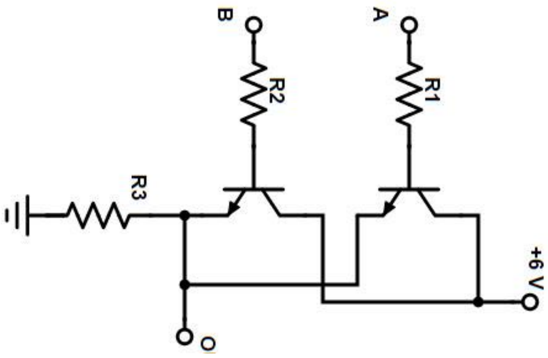


IEC Symbol



DIN Symbol

(CIRCUIT DIAGRAM OF OR GATE USING TRANSISTORS)

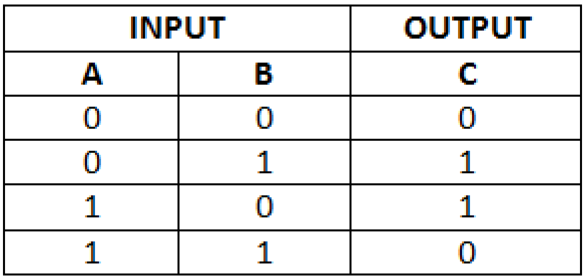


IC FOR OR GATE: 74(LS/HC)32

XOR GATE

The XOR gate (sometimes EOR gate, or EXOR gate and pronounced as Exclusive OR gate) is a digital logic gate that implements an exclusive or; that is, a true output (1/HIGH) results if one, and only one, of the inputs tothe gate is true. If both inputs are false (0/LOW) or both are true, a false output results. XOR represents the inequality function, i.e., the output is true if the inputs are not alike otherwise the output is false. A way to remember XOR is "one or the other but not both".

XOR can also be viewed as addition modulo 2. As a result, XOR gates are used to implement binary addition in computers. A Half adder consists of an XOR gate and an AND gate. Other uses include subtractors, comparators, and controlled inverters.



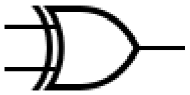
INPUT		OUTPUT
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

Truth Table

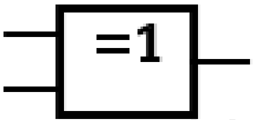
The algebraic expressions A.B’ + B.A’ and (A+B).(A’+B’) both represent the XOR gate with inputs A and B. The behavior of XOR is summarized in the truth table shown on the right.

SYMBOLS

There are two symbols for XOR gates: the traditional symbol and the IEEE symbol



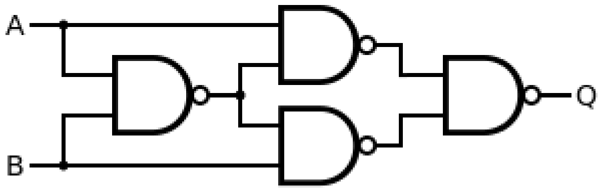
Traditional XOR Symbol



IEEE XOR Symbol

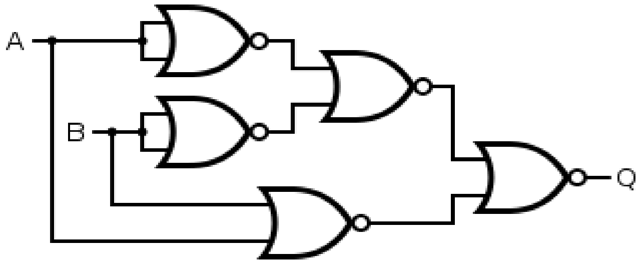
ALTERNATIVES

If a specific type of gate is not available, a circuit that implements the same function can be constructed from other available gates. A circuit implementing an XOR function can be trivially constructed from an XNOR gate followed by a NOT gate. If we consider the expression A.B'+A'.B , we can construct an XOR gate circuit directly using AND, OR and NOT gates. An XOR gate circuit can also be made from four NAND or five NOR gates in the configuration shown below.



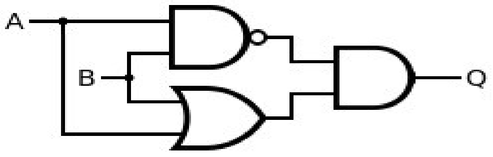
XOR gate circuit constructed using only only

NAND gates.



XOR gate circuit constructed using

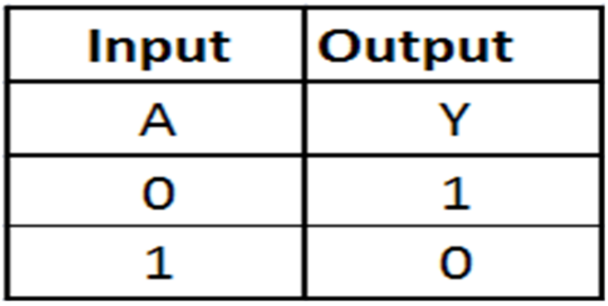
NOR gates.



XOR gate circuit using threemixed gates.

NOT GATE

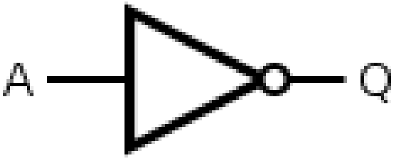
A NOT gate (also often called Inverter) is a logic gate. Each NOT gate has only one input signal. Logically with NOT gates, the input and the output swap, so if you input 1 it outputs as 0; likewise if you input 0 it outputs as 1.The NOT gate negates the values of data or signal in its input. It will always output the opposite signal. Its main function is to interchange logic.



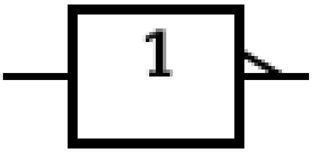
Input	Output
A	Y
0	1
1	0

SYMBOLS

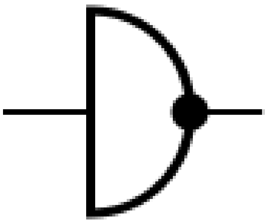
There are three symbols for the NOT gate:



MIL/ANSI Symbol

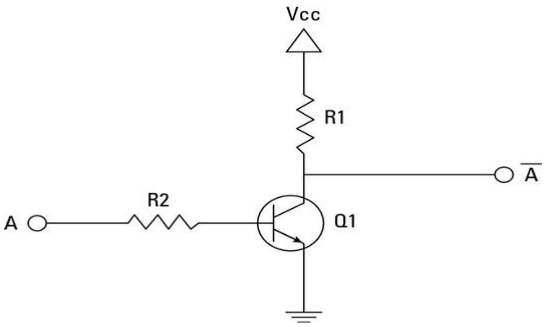


IEC Symbol



DIN Symbol

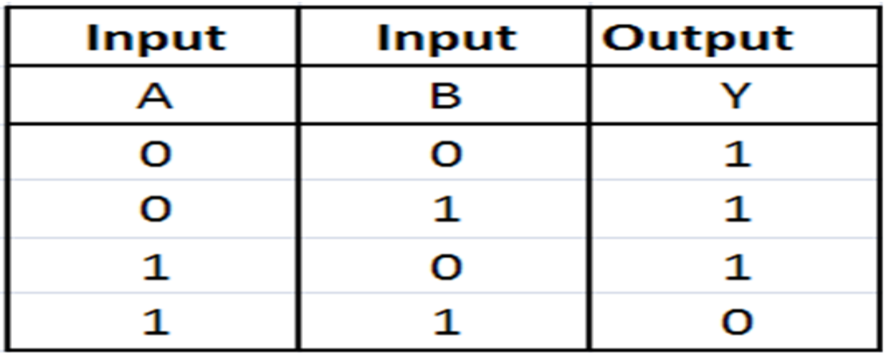
(CIRCUIT DIAGRAM OF NOT GATE USING TRANSISTORS)



IC FOR NOT GATE: 74(LS/HC)04

NAND GATE

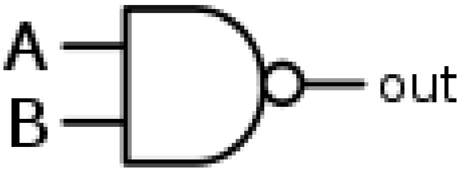
A NAND gate is a logic gate used to build digital logic circuits. It is a combination of an AND and NOT gate. The name refers to this. The NAND gate is a “universal gate”, that means all other types of logic gates can be obtained by wiring exclusively one or more NAND gates in a particular manner.



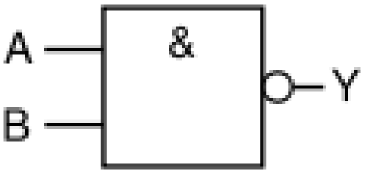
Input	Input	Output
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

SYMBOLS

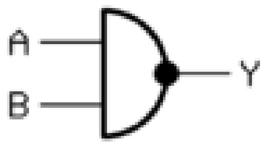
There are three symbols for drawing NAND gates in electrical circuit schematics: the “distinctive shape” symbol, the “military” symbol, and the “rectangular” symbol. For more information see logic gate symbols.



MIL/ANSI Symbol

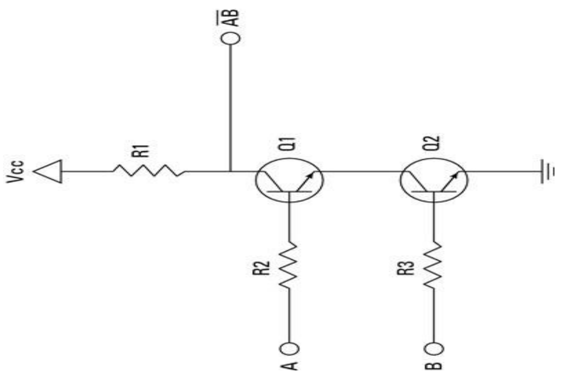


IEC Symbol



DIN Symbol

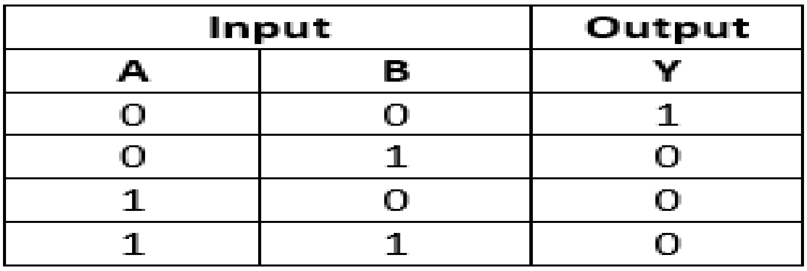
(CIRCUIT DIAGRAM OF NAND GATE USING TRANSISTORS)



IC FOR NAND GATE: 74(LS/HC)00

NOR GATE

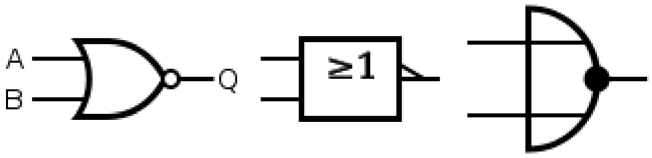
The NOR gate is a logic gate that outputs 1 (true) when both of its inputs are 0 (false). That means that if at least one of its inputs is 1 (true), the output will be 0. The best way to remember a NOR operation is “neither one or the other, nor both.”



Input		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

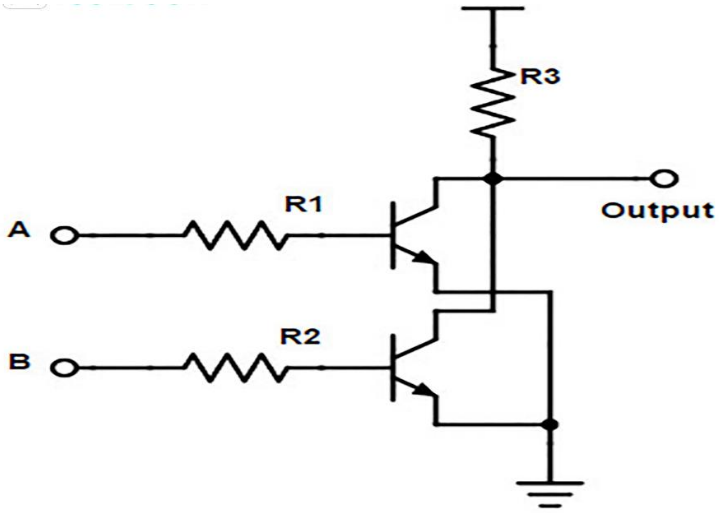
SYMBOLS

There are three symbols for the NOR gate:



MIL/ANSI Symbol *IEC Symbol* *DIN Symbol*

(CIRCUIT DIAGRAM OF NOR GATE USING TRANSISTORS)

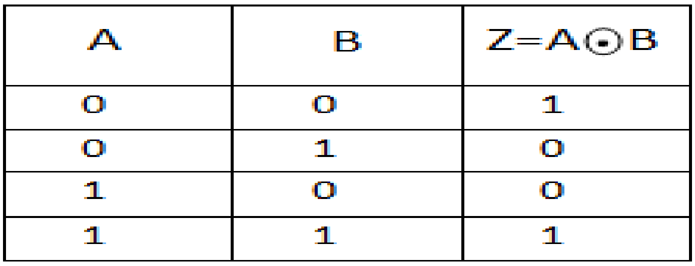


IC FOR NOR GATE: 74(LS/HC)02

XNOR GATE

An XNOR Gate is a type of digital logic gate that receives two inputs and produces one output. Both inputs are treated with the same logic, responding equally to similar inputs. Sometimes referred to as an “Equivalence Gate,” the gate’s output requires both inputs to be the same to produce a high output.

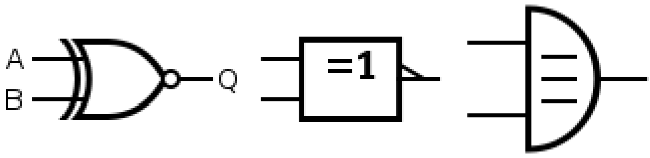
The gate works by receiving two inputs, each designated with either a 1 or a 0. If both inputs are 0, the gate will produce a 1. If both inputs are 1, the gate will also produce a 1. However, if either input differs from the other, the gate will output a 0.



A	B	$Z=A\odot B$
0	0	1
0	1	0
1	0	0
1	1	1

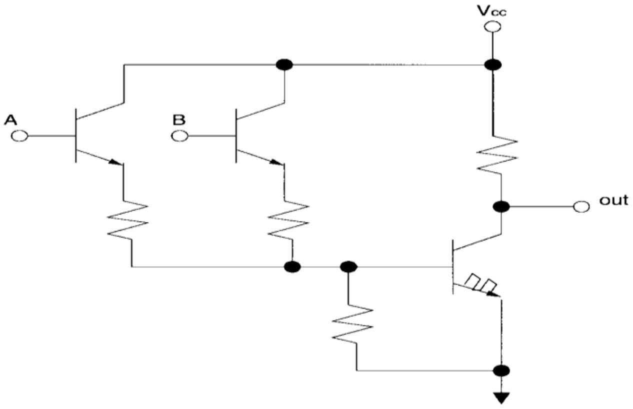
SYMBOLS

There are three symbols for the XNOR gate:



MIL/ANSI Symbol IEC Symbol DIN Symbol

(CIRCUIT DIAGRAM OF XNOR GATE USING TRANSISTORS)



IC FOR XNOR GATE:74(LS/HC)266

ADDER

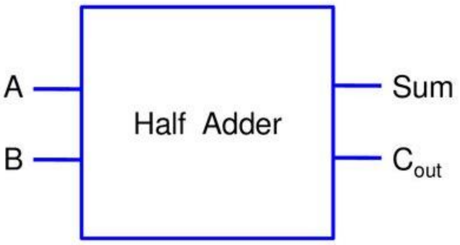
In electronics, an adder or summer is a digital circuit that performs addition of numbers. Although adders can be constructed for many numerical representations, such as binary-coded decimal or excess-3, the most common adders operate on binary numbers. In cases where two's complement or ones' complement is being used to represent negative numbers, it is trivial to modify an adder into an adder– subtractor. Other signed number representations require more logic around the basic adder.

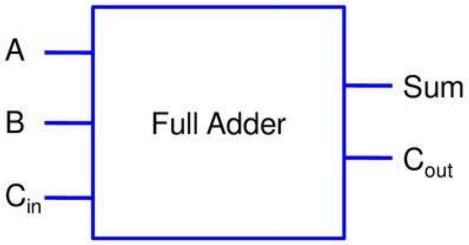
An adder can be used as a part of many other larger circuits like:

1) Ripple carry adder, it adds n-bits at a time.
2) Carryout Multiplication –the dedicated multiplication circuit uses it.
3) To generate memory addresses inside a computer and to make the Program Counter point to next instruction, the ALU makes use of this adder.
4) For graphics related applications, where there is a very much need of complex computations, the GPU uses optimized ALU which is made up of full adders, other circuits as well….
5) Basically, it is used in designing ALU and this ALU is used for wide variety of applications (from designing CPU to GPU).

There are two kinds of adders:

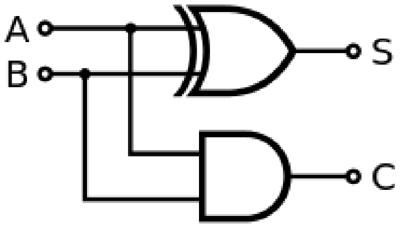
1) Half Adder:
   A half adder just adds two bits together and gives a two-bit output.
2) Full Adder:
   A full adder adds two inputs and a carried input from another adder, and also gives a two-bit output.





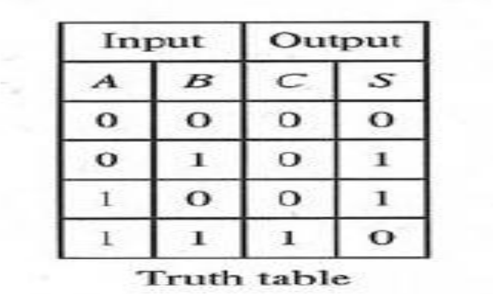
HALF ADDER

The half adder adds two single binary digits A and B. It has two outputs, sum (S) and carry (C). The carry signal represents an overflow into the next digit of a multi-digit addition. The value of the sum is 2C + S. The simplest half-adder design, pictured on the right, incorporates an XOR gate for S and an AND gate for C. With the addition of an OR gate to combine their carry outputs, two half adders can be combined to make a full adder.



Half adder logic diagram

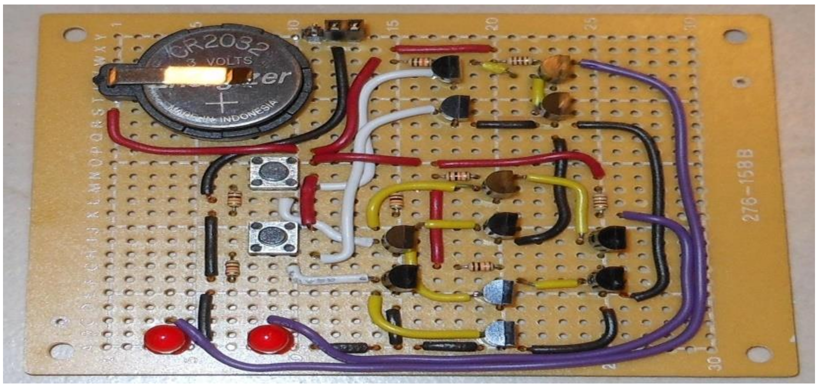
The half adder adds two input bits and generates a carry and sum, whichare the two outputs of a half adder. The input variables of a half adder arecalled the augend and addend bits. The output variables are the sum andcarry. The truth table for the half adder is:



Input		Output	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

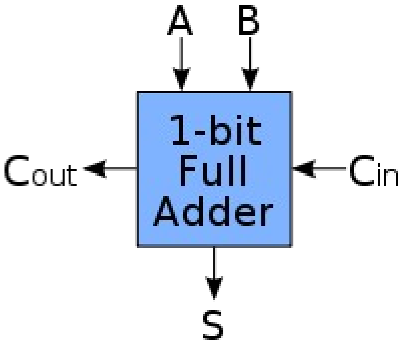
Truth table

(IMAGE OF HALF BIT ADDER USING TRANSISTORS)

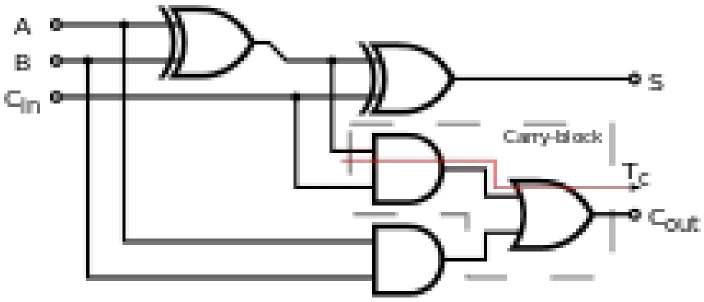


FULL ADDER

A full adder adds binary numbers and accounts for values carried in as wellas out. A one-bit full adder adds three one-bit numbers, often written as *A*,*B*, and C_{in}; *A* and *B* are the operands, and C_{in} is a bit carried in from the previous less significant stage. The full adder is usually a component in a cascade of adders, which add 8, 16, 32, etc. bit binary numbers. The circuit produces a two-bit output, output carry and sum typically represented by the signals C_{out} and *S*, where sum = $2C_{\text{out}}$ + S.



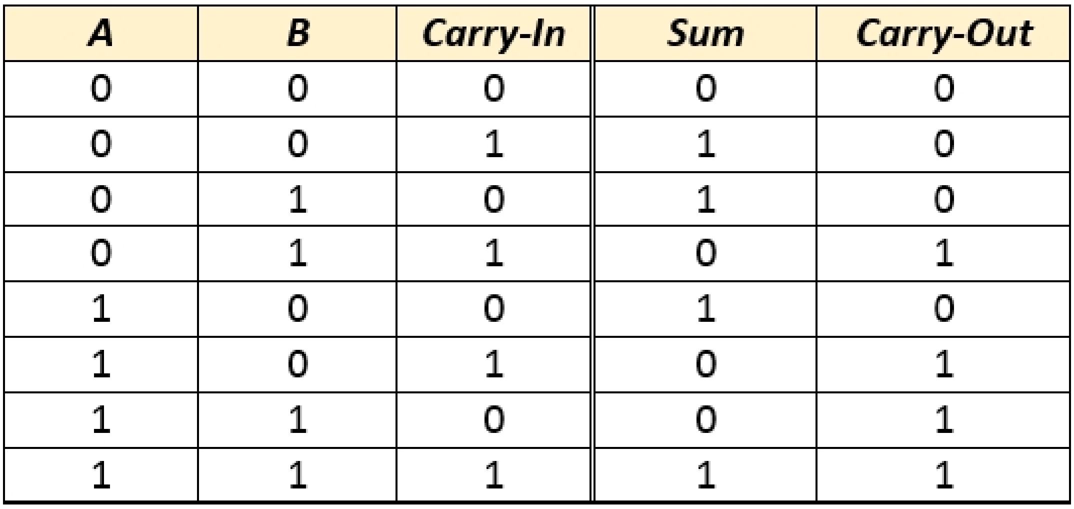
Schematic symbol for a 1-bit full adder.



Logic diagram for a full adder.

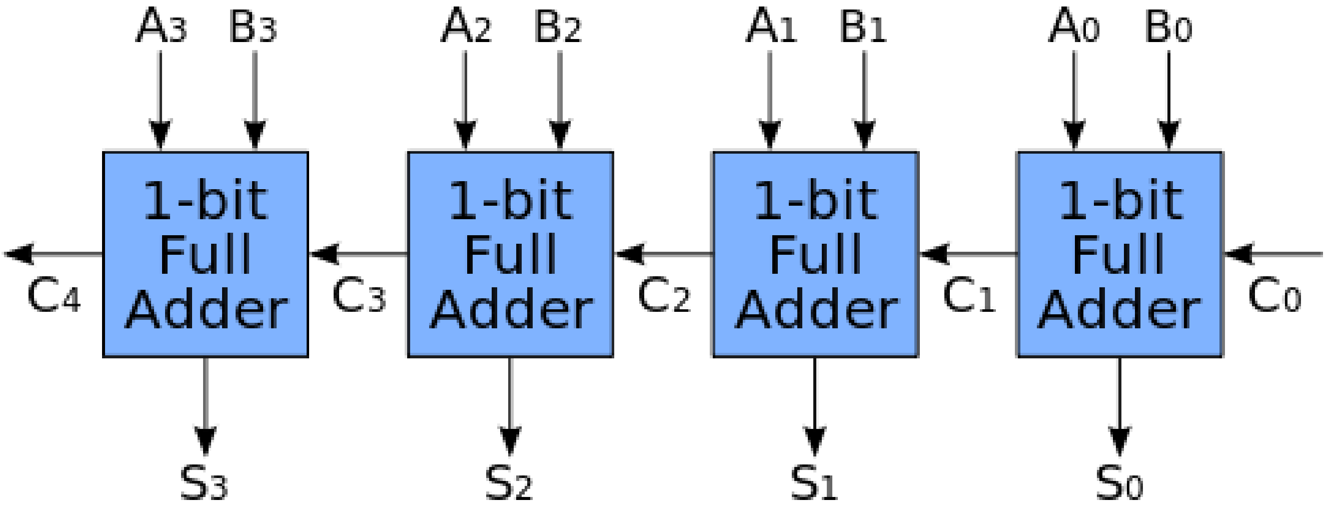
In this implementation, the final OR gate before the carry-out output maybe replaced by an XOR gate without altering the resulting logic. Using onlytwo types of gates is convenient if the circuit is being implemented using simple IC chips which contain only one gate type per chip.

A full adder can be constructed from two half adders by connecting *A* and *B* to the input of one half adder, connecting the sum from that to an inputto the second adder, connecting C_i to the other input and OR the two carryoutputs.



A	B	Carry-In	Sum	Carry-Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

RIPPLE CARRY ADDER

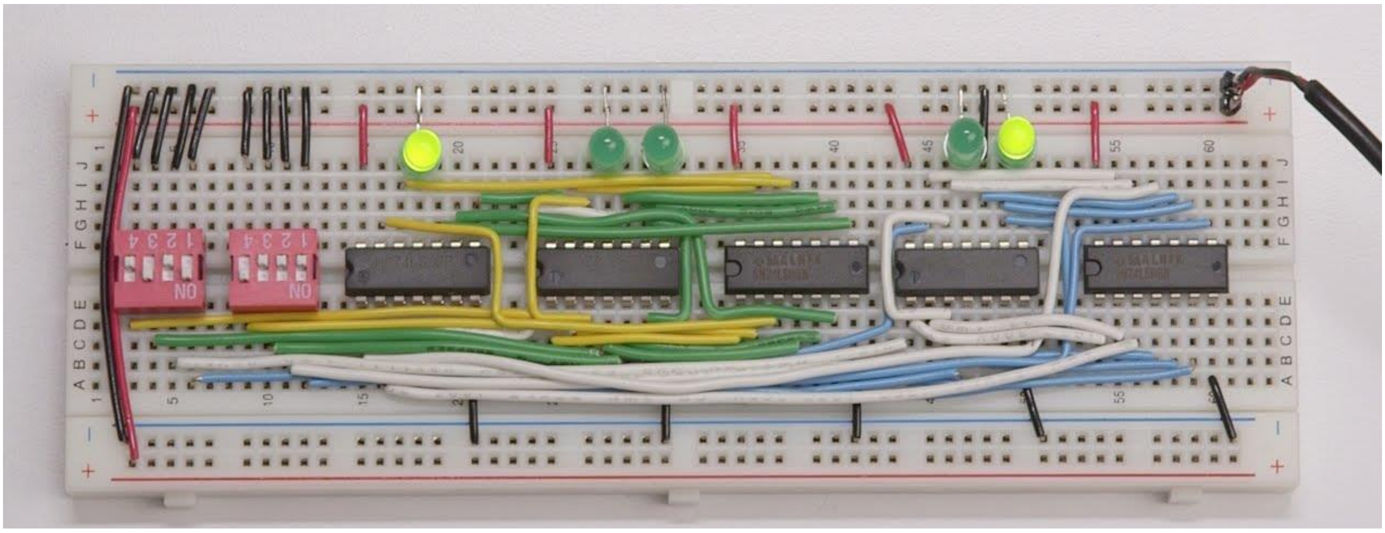


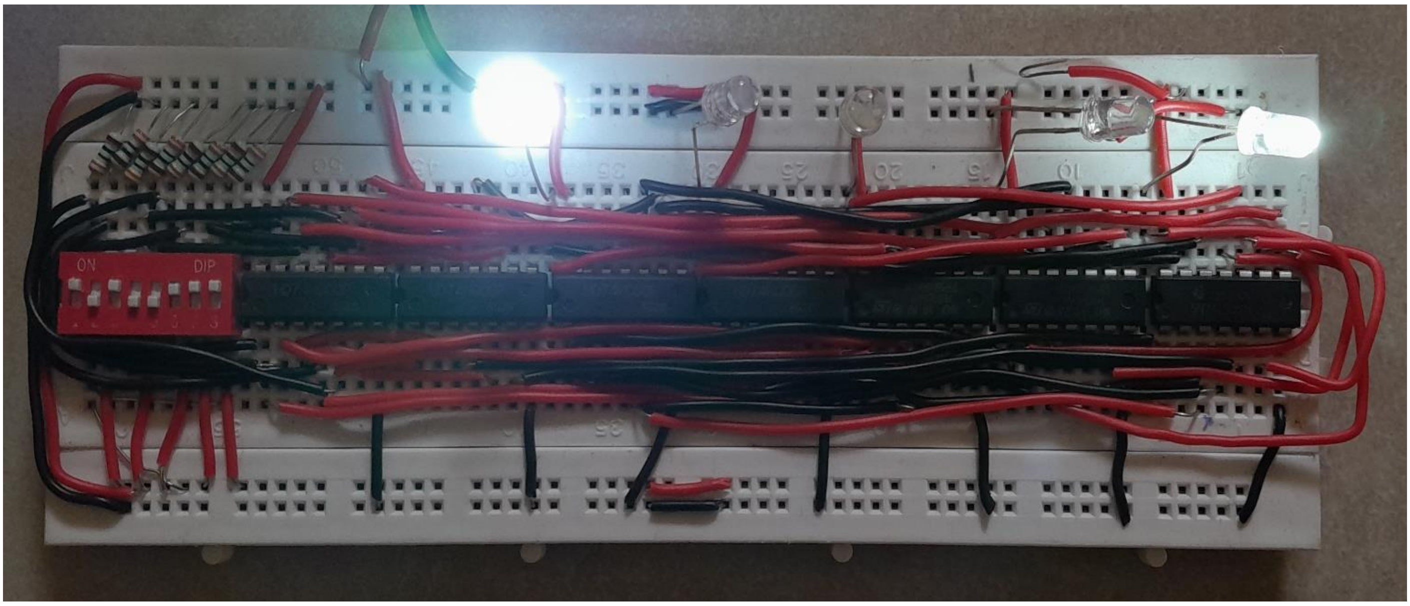
It is possible to create a logical circuit using multiple full adders to add *N*-bit numbers. Each full adder inputs a C_{in}, which is the C_{out} of the previous adder. This kind of adder is called a *ripple-carry adder*, since each carry bit "ripples" to the next full adder. Note that the first full adder may be replaced by a half adder (under the assumption that $C_{in} = 0$).

The layout of a ripple-carry adder is simple, which allows for fast design time; however, the ripple-carry adder is relatively slow, since each full adder must wait for the carry bit to be calculated from the previous full adder. The gate delay can easily be calculated by inspection of the full adder circuit. Each full adder requires three levels of logic.

In a 4-bit ripple-carry adder, there are 4 full adders, so the critical path (worst case) delay is 3 (from input to carry in first adder) + 3 * 2 (for carry propagation in later adders) = 9 gate delays.

(IMAGE OF FOUR BIT ADDER USING LOGIC GATE ICS)





BIBLIOGRAPHY

WEBSITES

- www.wikipedia.com
- www.techtarget.com
- www.researchgate.net
- www.quora.com
- www.dummies.com
- www.wikibooks.org
- www.electrical4u.com
- www.youtube.com/c/BenEater

BOOKS

- **NCERT Physics Book (Class-XII)**
- **Morris Mano Architecture**

ADDER

In electronics, an adder or summer is a digital circuit that performs addition of numbers. Although adders can be constructed for many numerical representations, such as binary-coded decimal or excess-3, the most common adders operate on binary numbers. In cases where two's complement or ones' complement is being used to represent negative numbers, it is trivial to modify an adder into an adder– subtractor. Other signed number representations require more logic around the basic adder.

An adder can be used as a part of many other larger circuits like:

6) Ripple carry adder, it adds n-bits at a time.
7) Carryout Multiplication –the dedicated multiplication circuit uses it.
8) To generate memory addresses inside a computer and to make the Program Counter point to next instruction, the ALU makes use of this adder.
9) For graphics related applications, where there is a very much need of complex computations, the GPU uses optimized ALU which is made up of full adders, other circuits as well….
10) Basically, it is used in designing ALU and this ALU is used for wide variety of applications (from designing CPU to GPU).

There are two kinds of adders:

3) Half Adder:

A half adder just adds two bits together and gives a two-bit output.

4) Full Adder:

A full adder adds two inputs and a carried input from another adder, and also gives a two-bit output.