Combinational Logic Circuits

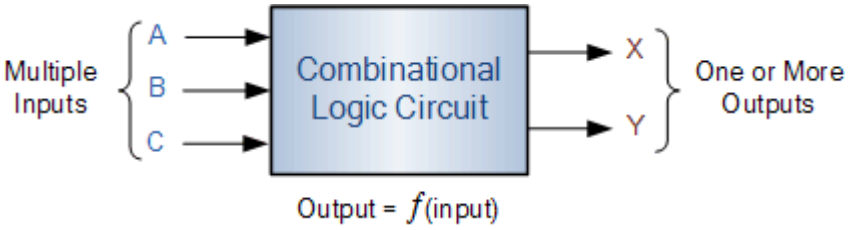
Combinational Logic Circuits are memoryless digital logic circuits whose output at any instant in time depends only on the combination of its inputs.

Unlike Sequential Logic Circuits whose outputs are dependant on both their present inputs and their previous output state giving them some form of *Memory*. The outputs of **Combinational Logic Circuits** are only determined by the logical function of their current input state, logic "0" or logic "1", at any given instant in time.

The result is that combinational logic circuits have no feedback, and any changes to the signals being applied to their inputs will immediately have an effect at the output. In other words, in a **Combinational Logic Circuit**, the output is dependant at all times on the combination of its inputs. Thus a combinational circuit is *memoryless*.

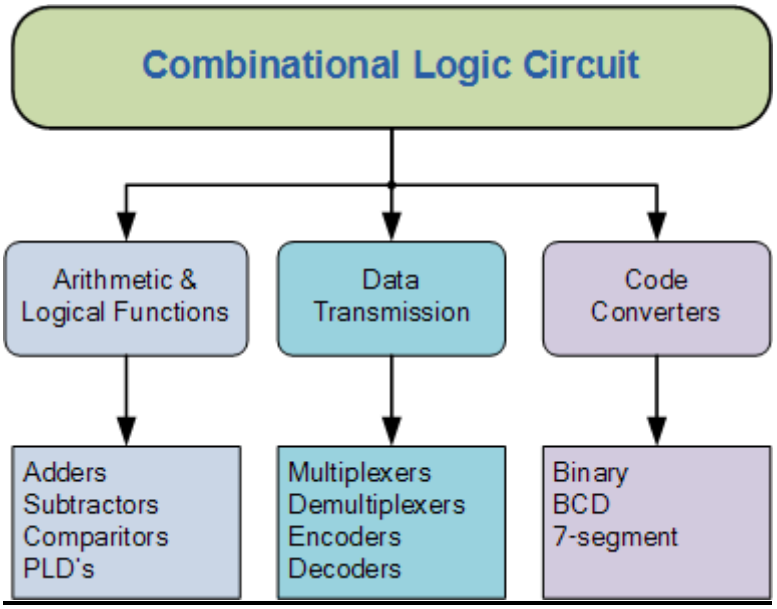
So if one of its inputs condition changes state, from 0-1 or 1-0, so too will the resulting output as by default combinational logic circuits have "no memory", "timing" or "feedback loops" within their design.

Combinational Logic



Common combinational circuits made up from individual logic gates that carry out a desired application include *Multiplexers*, *De-multiplexers*, *Encoders*, *Decoders*, *Full* and *Half Adders* etc.

Classification of Combinational Logic



Binary Adders

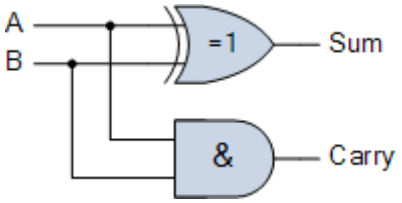
A common and very useful combinational logic circuit which can be constructed using just a few basic logic gates allowing it to add together two or more binary numbers is the **Binary Adder**.

123	A	(Augend)
+ 789	B	(Addend)
912	SUM	

Binary Adders are arithmetic circuits in the form of half-adders and full-adders used to add together two binary digits.

1. **A Half Adder Circuit**

A half adder is a logical circuit that performs an addition operation on two binary digits. The half adder produces **a sum and a carry value** which are both binary digits.

Logic circuit	Truth Table			
	A	B	SUM	CARRY
	0	0	0	0
	0	1	1	0
	1	0	1	0
	1	1	0	1

For the **SUM** bit:

$$SUM = A \text{ XOR } B = A \oplus B$$

For the **CARRY** bit:

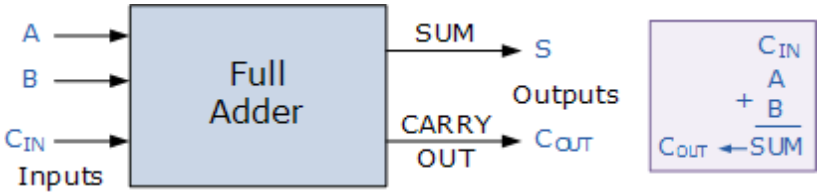
$$CARRY = A \text{ AND } B = A.B$$

From the truth table of the half adder we can see that the SUM (S) output is the result of the Exclusive-OR gate and the Carry-out (Cout) is the result of the AND gate. Then the Boolean expression for a half adder is as above.

2. **A Full Adder Circuit**

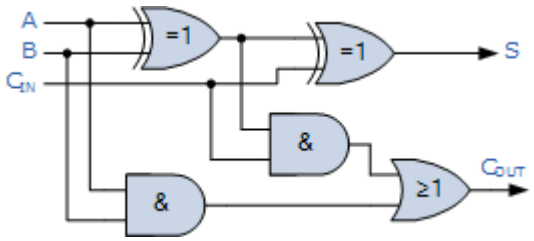
The main difference between the **Full Adder** and the previous **Half Adder** is that a full adder has three inputs. The same two single bit data inputs A and B as before plus an additional *Carry-in* (C-in) input to receive the carry from a previous stage as shown below.

Full Adder Block Diagram



Then the **full adder** is a logical circuit that performs an addition operation on three binary digits and just like the half adder, it also generates a carry out to the next addition column. Then a *Carry-in* is a possible carry from a less significant digit, while a *Carry-out* represents a carry to a more significant digit.

Full Adder Truth Table with Carry

Logic circuit	Truth Table				
	A	B	C-in	Sum	C-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

Then the Boolean expression for a full adder is as follows.

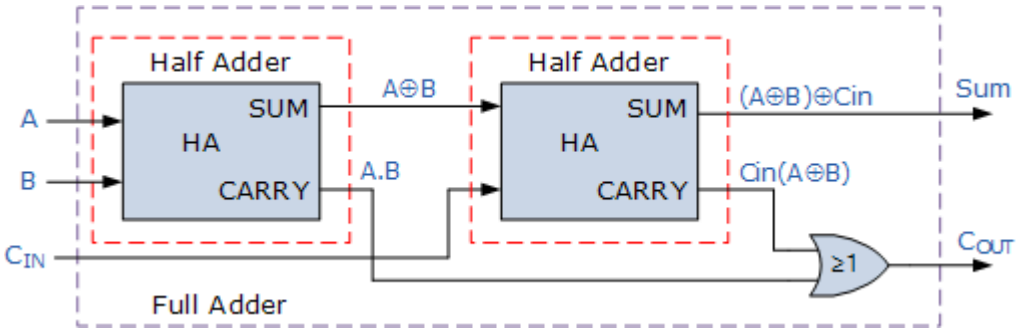
For the **SUM** (S) bit:

$$SUM = (A \text{ XOR } B) \text{ XOR } Cin = (A \oplus B) \oplus Cin$$

For the **CARRY-OUT** (Cout) bit:

$$CARRY\text{-}OUT = A \text{ AND } B \text{ OR } Cin(A \text{ XOR } B) = A.B + Cin(A \oplus B)$$

Full Binary Adder Logic Diagram



As the full adder circuit above is basically two half adders connected together, the truth table for the full adder includes an additional column to take into account the *Carry-in*, C_{IN} input as well as the summed output, S and the Carry-out, C_{OUT} bit.

An n-bit Binary Adder

We have seen above that single 1-bit binary adders can be constructed from basic logic gates. But what if we wanted to add together two n-bit numbers, then n number of 1-bit full adders need to be connected or "cascaded" together to produce what is known as a **Ripple Carry Adder**.

A "ripple carry adder" is simply "n", 1-bit full adders cascaded together with each full adder representing a single weighted column in a long binary addition. It is called a ripple carry adder because the carry signals produce a "ripple" effect through the binary adder from right to left, (LSB to MSB).

For example, suppose we want to "add" together two 4-bit numbers, the two outputs of the first full adder will provide the first place digit sum (S) of the addition plus a carry-out bit that acts as the carry-in digit of the next binary adder.

A 4-bit Ripple Carry Binary Adder