Contents

Combinational logic circuits 2
Example 1. Combinational Logic Circuit 3
Classification of Combinational Logic 3
Example 2. Combinational Logic Circuit 4
Example 3. Combinational Logic Circuit – HALF ADDERS 4
Example 4. Combinational Logic Circuit – FULL ADDER 6
More Complex Adder Topologies 7

Figure 1: Classifications of Combinational Logic Circuits 3
Figure 2: Half-Adder Circuit Diagram 5
Figure 3: Hald Adder Truth Table 5
Figure 4: Full Adder Truth Table 6
Figure 5: Full adder Circuit Diagram 6
Figure 6: Full-adder represented as two half-adders 7

Combinational logic circuits

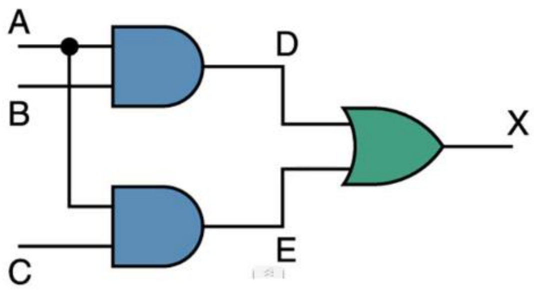
The ***combinational logic circuits*** or time-independent logic circuits in digital circuit theory can be defined as a type of digital logic circuit implemented using Boolean circuits, where the output of logic circuit is a pure function of the *present* inputs only.

The combinational logic circuit operation is instantaneous and these circuits do not have the memory or feedback loops.

This combinational logic is in contrast compared to the ***sequential logic circuit*** in which the output depends on both present inputs and also on the previous inputs.

Thus, we can say that combinational logic **<u>does not have memory</u>**, whereas sequential logic **<u>stores previous input in its memory</u>**. Hence, if the input of combinational logic circuit changes, then the output also changes.

Gates are combined into circuits by using the output of one gate as the input for another (recall: the input values explicitly determine the output)



Combinational logic circuits are generally designed by connecting together or combining the basic logic gates such as NAND, NOR, and NOT.

Hence, these logic gates are termed as building blocks. These logic circuits can be a very simple circuit or a very complex circuit or huge combinational circuit can be designed using only *universal logic gates* such as NAND and NOR gates.

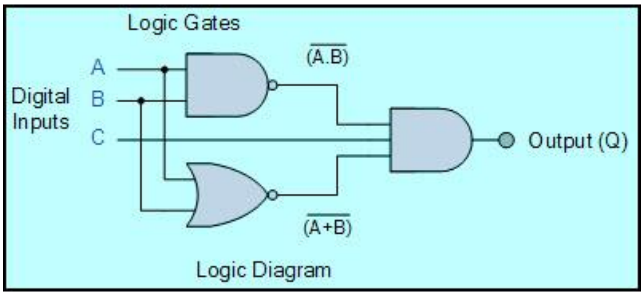
A	B	C	D	E	X
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

Example 1. Combinational Logic Circuit

(1) Boolean Expression

$$Q = \overline{(A.B)}.\overline{(A+B)}.C$$

(2) Logic Diagram



(3) Truth Table

C	B	A	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Classification of Combinational Logic

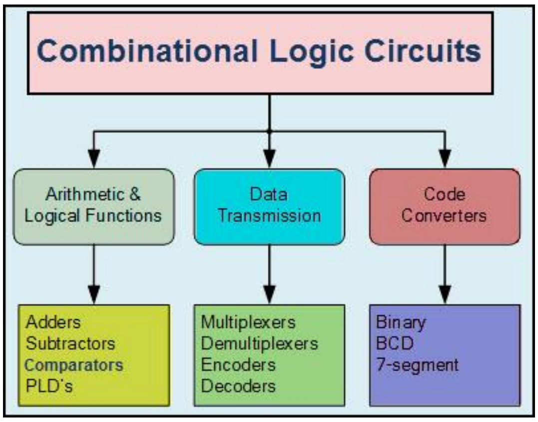


Figure 1: Classifications of Combinational Logic Circuits

The combinational logic circuits can be classified into various types based on the purpose of usage, such as

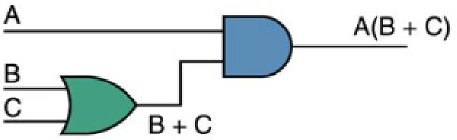
- arithmetic & logical functions,
- data transmission, and
- code converters.

To solve the arithmetic and logical functions we generally use **adders**, subtractors, and comparators which are generally realised by combining various logic gates called as combinational logic circuits.

Similarly, for data transmission, we use **multiplexers**, demultiplexers, encoders, and decoders which are also realised using combinational logic.

In fact, combinational logic is *most frequently used* in multiplexer and demultiplexer type circuits.

Example 2. Combinational Logic Circuit



A	B	C	B + C	A(B+C)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

There are four fundamental circuits of a CPU, an adder, a decoder, a shifter and flip-flops(latches)

These are all constructed from basic gates to do specific functions

Example 3. Combinational Logic Circuit – HALF ADDERS

At the digital logic level, addition is performed in binary

Addition operations are carried out by what are called “**adders**”

Function: add two binary numbers, output result

- Inputs: two bits (x,y) to add and one carry-in(C_{in})
- Outputs: sum bit (s) and one carry out but (C_{out})

The result of adding two binary digits could produce a carry value:

Binary Values	**Result when Added**
0+0	0
0+1	1
1+0	1
1+1	**10**

!!!!!!!!!! **$1_2+1_2 = 10_2$** **!!!!!!!!!!!!**

A circuit that computes the sum of two bits and produces the correct carry bit is called a half adder

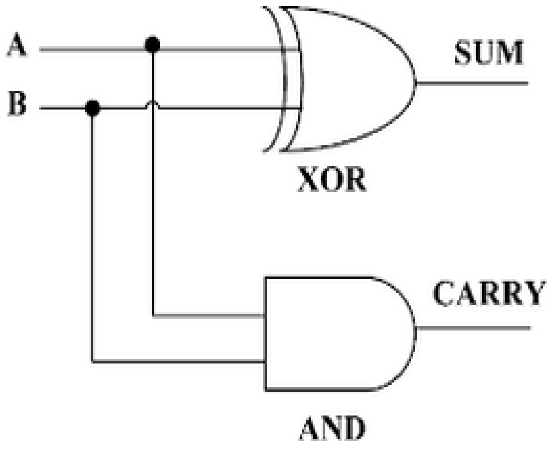


Figure 2: Half-Adder Circuit Diagram

A	**B**	**SUM ($A \oplus B$)**	**CARRY (AB)**

Figure 3: Hald Adder Truth Table

Recall: XOR – 1 when inputs are not the same; AND is A.B multiplied

Example 4. Combinational Logic Circuit – FULL ADDER

Full adder is developed to overcome the drawback of Half Adder circuit. It can add two one-bit numbers A and B, and carry c. The full adder is a three input and two output combinational circuit.

- A *full adder* can add a bit carried from another addition as well as the two inputs i.e. 3inputs
- A *half adder* can only add the inputs together.
    - i.e. A circuit called a full adder takes the carry-in value into account

Cin	A	B	Sum	Cout
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

Figure 4: Full Adder Truth Table

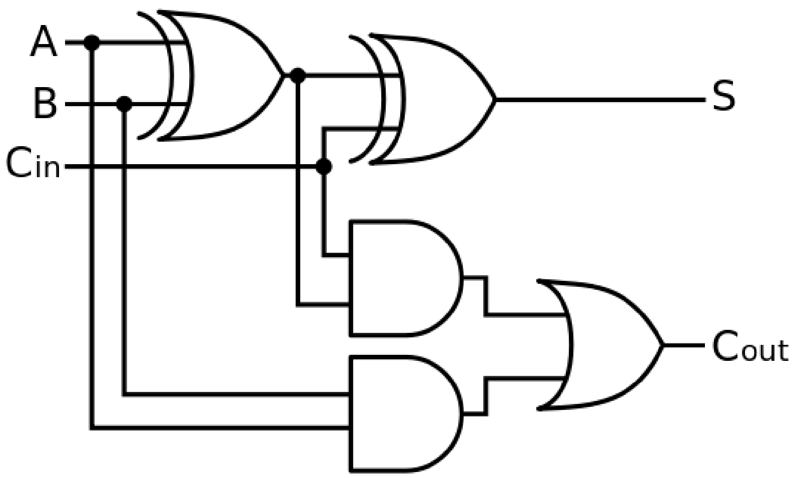


Figure 5: Full adder Circuit Diagram

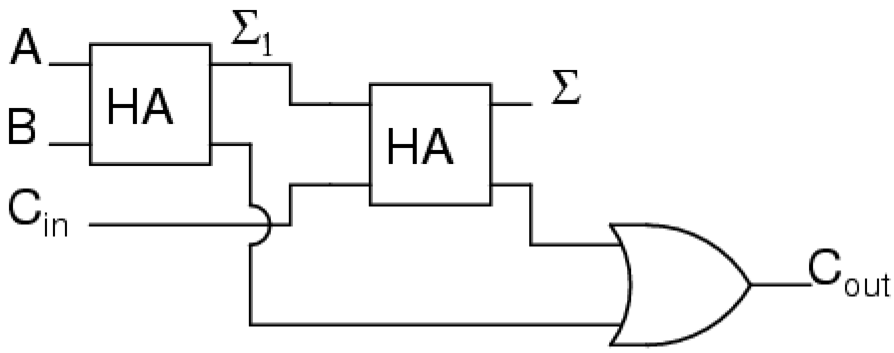


Figure 6: Full-adder represented as two half-adders

More Complex Adder Topologies

Adders become a critical hardware unit for the efficient implementation of arithmetic unit. In many arithmetic applications and other kinds of applications, adders are not only in the arithmetic logic unit, but also in other parts of processor. Addition operation can also be used in complement operations (1's, 2's, and so on), encoding, decoding and so on.

The adder we just created is called a **ripple-carry** adder. It gets that name because the carry bits "ripple" from one adder to the next.

This implementation has the advantage of **simplicity** but the disadvantage of **speed problems**.

> In a real circuit, gates take time to switch states (the time is on the order of nanoseconds, but in high-speed computers nanoseconds matter). So 32-bit or 64-bit ripple-carry adders might take 100 to 200 nanoseconds to settle into their final sum because of carry ripple.

For this reason, engineers have created more advanced adders, such as: with the design of various adders

- Ripple Carry Adder (RCA),
- Carry Skip Adder (CSkA),
- Carry Increment Adder (CIA),
- Carry Look Ahead Adder (CLA),
- Carry Save Adder (CSA),
- Carry Select Adder (CSlA) and
- Carry Bypass Adder (CBA)

Every adder above is named based on the **propagation of carry between the stages**.

Addition is an indispensable operation for any high speed digital system, digital signal processing or control system.

Therefore appropriate choice of adder topologies is an essential importance in the design of GSI (giga-scale integration) circuits for high speed and high performance circuits

All complex adder architectures are constructed from its basic building blocks such as Half Adder and Full Adder.