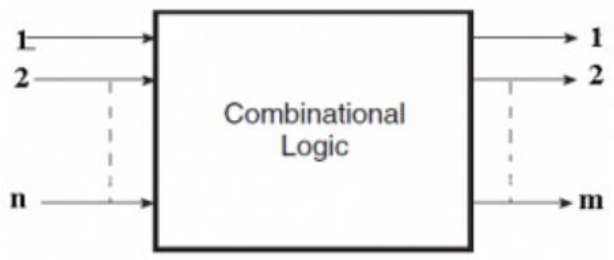
Combinational Circuits

- Combinational circuit is a circuit in which we combine the different gates in the circuit.
- Some of the characteristics of combinational circuits are following:
    - The output of combinational circuit at any instant of time, depends only on the levels present at input terminals.
    - The combinational circuit does not use any memory.
    - The previous state of input does not have any effect on the present state of the circuit.
    - A combinational circuit can have an n number of inputs and m number of outputs.

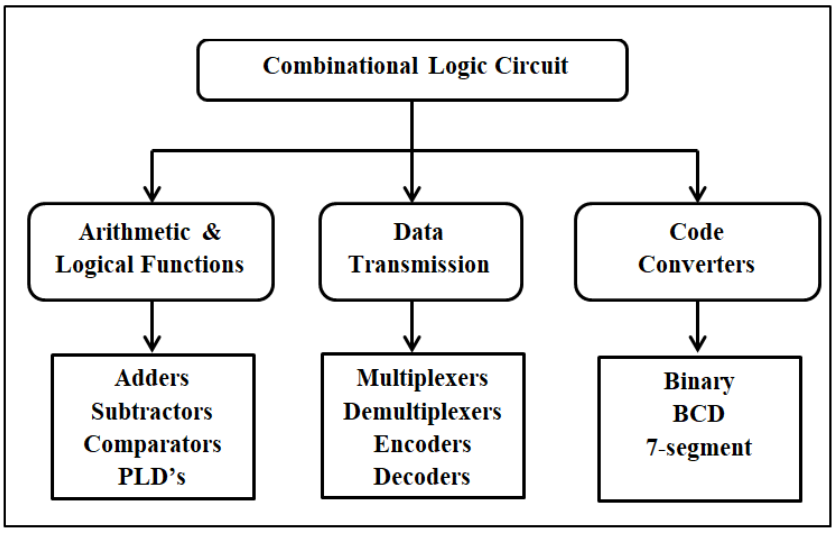
Block diagram



Block diagram of Combinational circuits

- As combinational logic circuits are made up from individual logic gates only, they can also be considered as "decision making circuits" and combinational logic is about combining logic gates together to process two or more signals in order to produce at least one output signal according to the logical function of each logic gate.
- 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**



We're going to elaborate few important combinational circuits as follows.

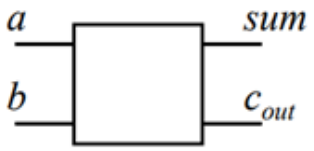
Arithmetic and Logic Function Digital circuits:

1- Adder

- **Half-Adder:**

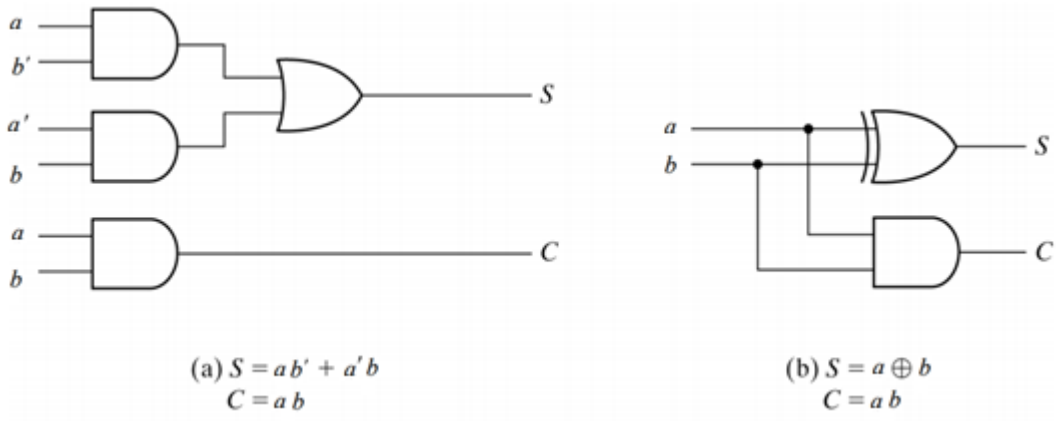
Half adder is a combinational logic circuit with two inputs and two outputs. The half adder circuit is designed to add two single bit binary number A and B. (note: no carry input).
It is the basic building block for addition of two single bit numbers.

a	b	sum	c_{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Truth Table and Block diagram of Half-Adder

This circuit has two outputs carry and sum.
The Boolean expressions for SUM and CARRY (C_{out}) are,
SUM = AB'+A'B = A ⊕ B
CARRY = AB
These expressions shows`that, SUM output is EX-OR gate and the CARRY output is AND gate. Figure below shows the implementation of half-adder with all the combinations including the implementation using NAND gates only.



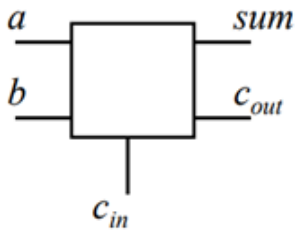
(a) $S = ab' + a'b$, $C = ab$ (b) $S = a \oplus b$, $C = ab$

Implementation of Half-Adder

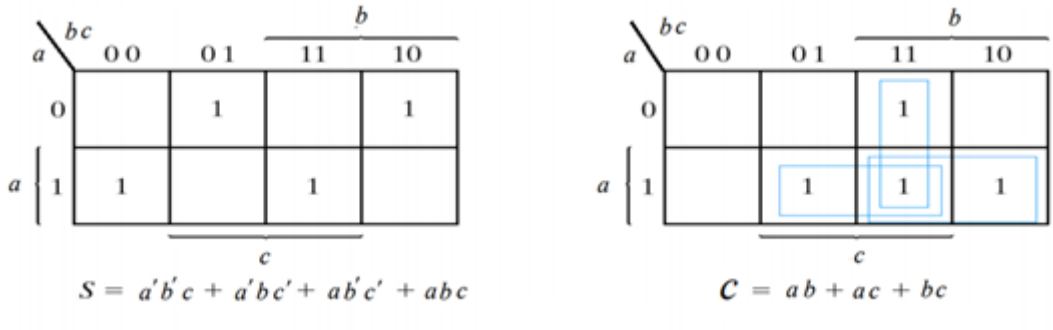
- **Full-Adder:**

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. (note: with a carry input). The full adder is a three input and two output combinational circuit.

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



Truth Table and Block diagram of Full-Adder

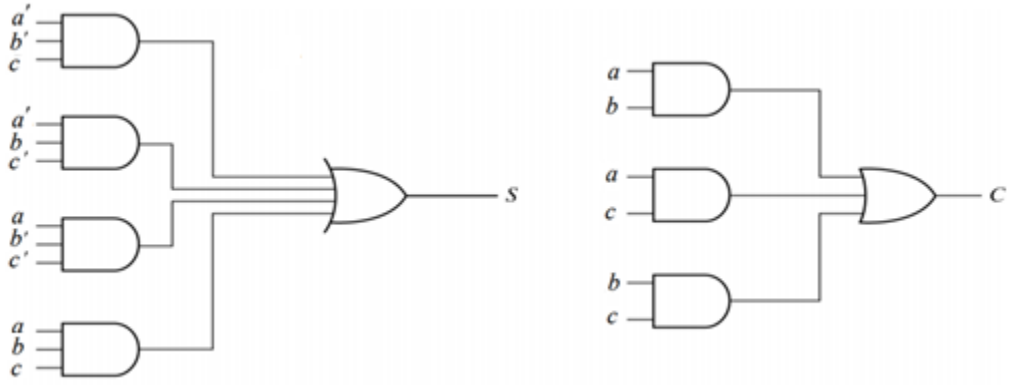


K-Map simplification for Full-Adder

S = a'b'c + a'bc' + ab'c' + abc

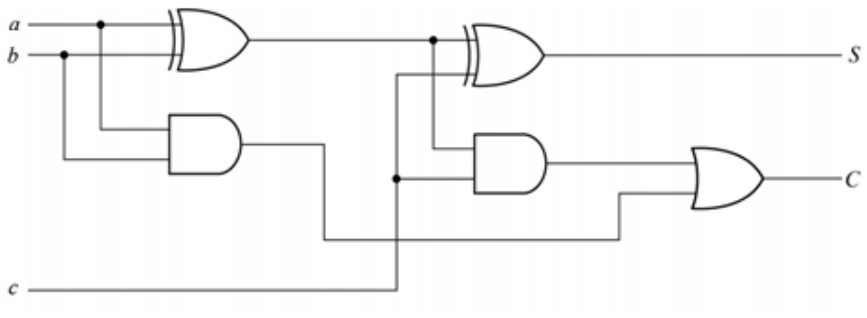
C = ab + ac + bc

Full adder implemented in SOP:



Implementation of Full-Adder in Sum of Products (SoP)

Another implementation:
Full-adder can also implemented with two half adders and one OR gate (Carry Look-Ahead adder).
$S = c \oplus (a \oplus b)$
$C = c\,(ab' + a'b) + ab = ab'c + a'bc + ab$



Implementation of Full-adder with two half adders and one OR gate

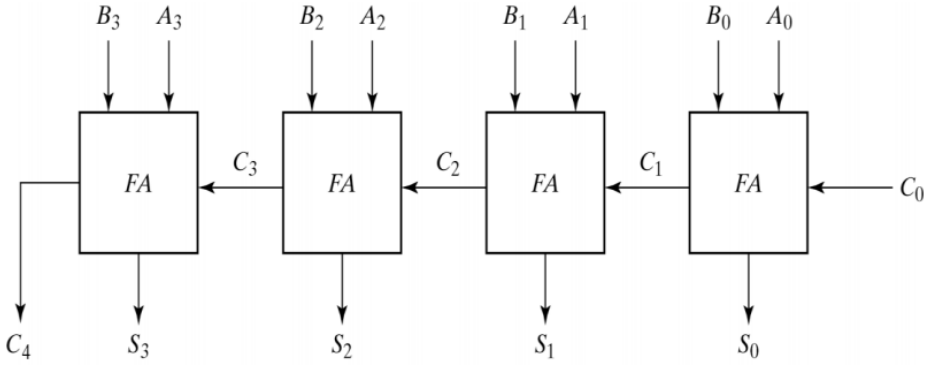
- **N-Bit Parallel Adder:**

The Full Adder is capable of adding only two single digit binary number along with a carry input. But in practical we need to add binary numbers which are much longer than just one bit.
To add two n-bit binary numbers we need to use the n-bit parallel adder. It uses a number of full adders in cascade. The carry output of the previous full adder is connected to carry input of the next full adder.

Four-Bit Parallel Adder

In the block diagram, A_0 and B_0 represent the LSB of the four bit words A and B. Hence Full Adder-0 is the lowest stage. Hence its C_{in} has been permanently made 0. The rest of the connections are exactly same as those of n-bit parallel adder is shown in fig. The four bit parallel adder is a very common logic circuit.

Block diagram



Four-bit parallel Adder

This is also called Ripple Carry Adder ,because of the construction with full adders are connected in cascade.

Subscript i:	3	2	1	0	
Input carry	0	1	1	0	C_i
Augend	1	0	1	1	A_i
Addend	0	0	1	1	B_i
Sum	1	1	1	0	S_i
Output carry	0	0	1	1	C_{i+1}

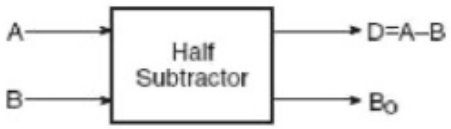
It causes an unstable factor on carry bit, and produces a longest propagation delay. The signal from C_i to the output carry C_{i+1}, propagates through an AND and OR gates, so, for an n-bit RCA, there are 2n gate levels for the carry to propagate from input to output. Because the propagation delay will affect the output signals on different time, so the signals are given enough time to get the precise and stable outputs.

2- Subtracter

- **Half-Subtractor:**

A half-subtractor is a combinational logic circuit which has two inputs and two outputs. Let the input variables minuend and subtrahend be designated as X and Y respectively, and output functions be designated as D for difference and B for borrow. The truth table of the functions is as follows.

A	B	D	B_O
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



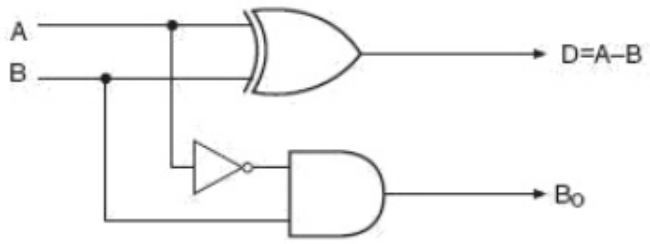
Truth Table and Block diagram of Half-Subtractor

By considering the minterms of the truth table above, the Boolean expressions of the outputs D and B functions can be written as

$D = A'B + AB'$ and

$B_o = A'B$.

Figure below shows the logic diagram to realize the half-subtractor circuit.



Implementation of Half-Subtractor

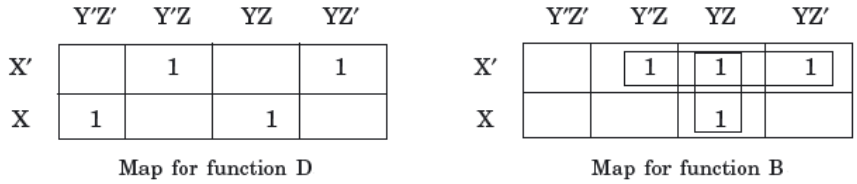
- **Full-subtractors**

A combinational circuit of full-subtractor performs the operation of subtraction of three bits—the minuend, subtrahend, and borrow generated from the subtraction operation of previous significant digits and produces the outputs difference and borrow.

Let us designate the input variables minuend as X, subtrahend as Y, and previous borrow as Z, and outputs difference as D and borrow as B. Eight different input combinations are possible for three input variables. The truth table is shown below according to its functions.

Input variables			Outputs	
X	Y	Z	D	B
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Truth Table of Full-Subtractor



K-Map simplification for Full- subtractor

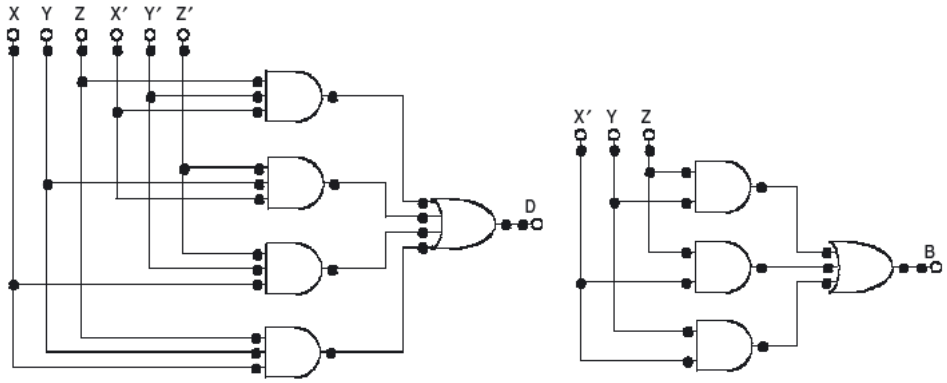
Karnaugh maps are prepared to derive simplified Boolean expressions of D and B as in Figures above.

The simplified Boolean expressions of the outputs are

D = X′Y′Z + X′YZ′ + XY′Z′ + XYZ and

B = X′Z + X′Y + YZ.

The logic diagram for the above functions is shown in Figure below:



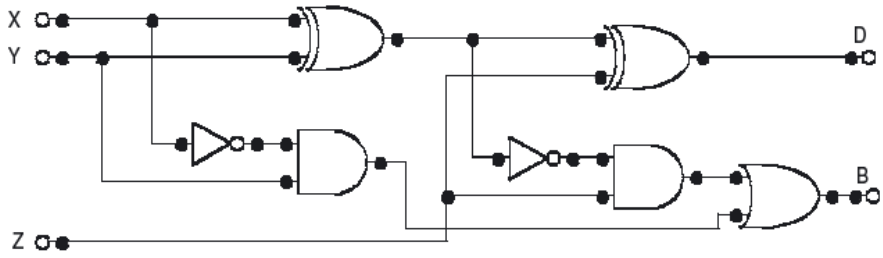
Implementation of Full-Subtractor

Similar to a full-adder circuit, it should be noticed that the configuration of the combinational circuit diagram for full-subtractor as shown in Figure above contains two-input and three-input AND gates, and three-input and four-input OR gates. Other configurations can also be developed where number and type of gates are reduced. For this, the Boolean expressions of D and B are modified as follows.

$D = X'Y'Z + X'YZ' + XY'Z' + XYZ$
$= X' (Y'Z + YZ') + X (Y'Z' + YZ)$
$= X' (Y \oplus Z) + X (Y \oplus Z)'$
$= X \oplus Y \oplus Z$

$B = X'Z + X'Y + YZ = X'Y + Z (X' + Y)$
$= X'Y + Z(X'Y + X'Y' + XY + X'Y)$
$= X'Y + Z(X'Y + X'Y' + XY)$
$= X'Y + X'YZ + Z(X'Y' + XY)$
$= X'Y + Z(X \oplus Y)'$

Logic diagram according to the modified expression is shown in Figure below.



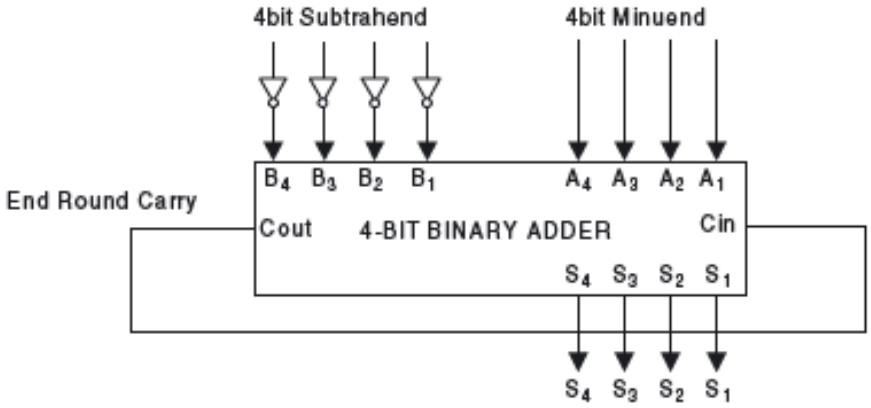
Implementation of Full-Subtractor

- **N-Bit Parallel Subtractor**

The subtraction can be carried out by taking the 1's or 2's complement of the number to be subtracted. For example we can perform the subtraction (A-B) by adding either 1's or 2's complement of B to A. That means we can use a binary adder to perform the binary subtraction.

Four-Bit Parallel Subtractor

It is interesting to note that a 4-bit binary adder can be employed to obtain the 4-bit binary subtraction. In lectures, we saw how binary subtraction can be achieved using 1's complement or 2's complement. By 1's complement method, the bits of subtrahend are complemented and added to the minuend. If any carry is generated it is added to the sum output. Figure below demonstrates the subtraction of $B_4B_3B_2B_1$ from $A_4A_3A_2A_1$. Each bit of $B_4B_3B_2B_1$ is first complemented by using INVERTER gates and added to $A_4A_3A_2A_1$ by a 4-bit binary adder. End round carry is again added using the C in pin of the IC.

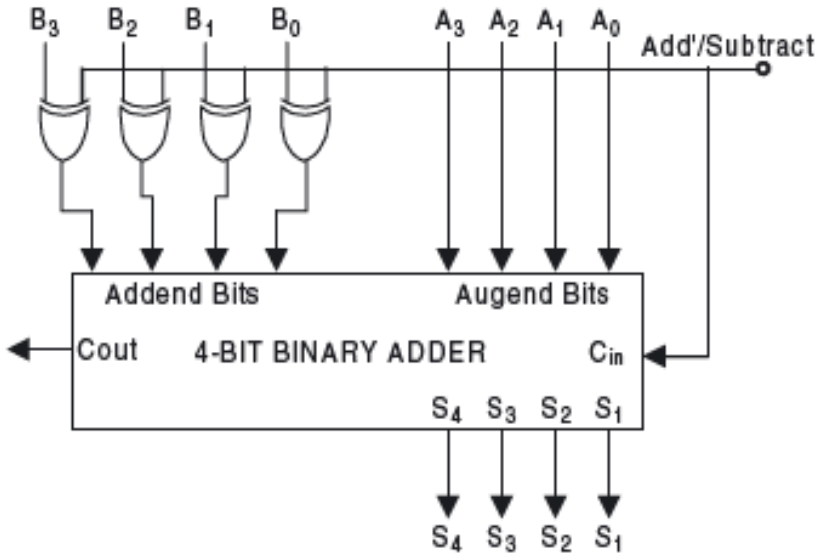


Block diagram of Four-Bit Parallel Subtractor

- **Four-bit Binary Parallel Adder/Subtractor**

Due to the property of the 4-bit binary adder that it can perform the subtraction operation with external inverter gates, a single combinational circuit may be developed that can perform addition as well as the subtraction introducing a control bit. A little modification helps to obtain this dual operation. Figure below demonstrates this dual-purpose combinational logic circuit.

XOR gates are used at addend or subtrahend bits when one of the inputs of the XOR gate is connected to the ADD/SUBTRACT terminal, which is acting as control terminal. The same terminal is connected to Cin. When this terminal is connected to logic 0 the combinational circuit behaves like a 4-bit full adder, as at this instant Cin is logic low and XOR gates are acting as buffers whose outputs are an uncomplemented form of inputs. If logic 1 is applied to the ADD/SUBTRACT terminal, the XOR gates behave like INVERTER gates and data bits are complemented. The 4-bit adder now performs the addition operation of data $A_3A_2A_1A_0$ with complemented form of data $B_3B_2B_1B_0$ as well as with a single bit 1, as Cin is now logic 1. This operation is identical to a subtraction operation using 2's complment.



Block diagram of Four-bit Binary Parallel Adder/Subtractor