COMBINATIONAL CIRCUIT

It is a circuit whose output depends upon present input at any instant of time. There is no memory element in this circuit. It can be designed using gates or available IC. Adder, subtractor, multiplexer, decoder, encoder and demultiplexer are example of combinational circuit.

BINARY ADDERS

An adder or summer is a digital circuit that performs addition of numbers. In many computers and other kinds of processors, adders are used not only in the arithmetic logic unit(s), but also in other parts of the processor, where they are used to calculate addresses, table indices, and similar. 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 one's complement is being used to represent negative numbers, it is trivial to modify an adder into an adder-subtractor. Other signed number representations require a more complex adder.

Half adder

It is used to add two binary numbers. In general, the adder circuit needs two binary inputs and two binary outputs. The input variables designate the augends and addend bits; the output variables produce the sum and carry.

The binary addition operation of single bit is shown in the truth table.

X	Y	C Carry bit	S Sum bit
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

The simplified sum of products expressions are:

$S=X'Y+ X Y'$

$C=XY$

The circuit implementation is as shown as given. This circuit cannot handle the carry input, so it is termed as half adder. The circuit diagram and block diagram of Half Adder is shown in Figure

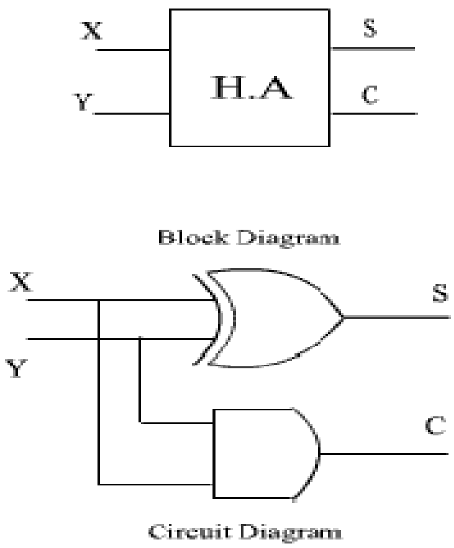


Figure: circuit and block diagram of half adder

Full Adder:

A full adder is a combinational circuit that forms the arithmetic sum of three bits. It consists of three inputs and two outputs. Two of the input variables, denoted by x and y, represent the two bits to be added. The third input Z, represents the carry from the previous lower position. The two outputs are designated by the symbols S for sum and C for carry.

X	Y	Z	C Carry bit	S Sum bit
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

The simplified expression for S and C are

$S=X'Y'Z+X'YZ'+XY'Z'+XYZ$

$=X'(Y'Z+YZ')+X(Y'Z'+YZ)$

$=X\oplus Y+\oplus Z$

$C=X'YZ+XY'Z+XYZ'+XYZ$

$=Z(X'Y+XY')+XY$

$=Z(X\oplus Y)+XY$

The circuit diagram full adder is shown in the figure.

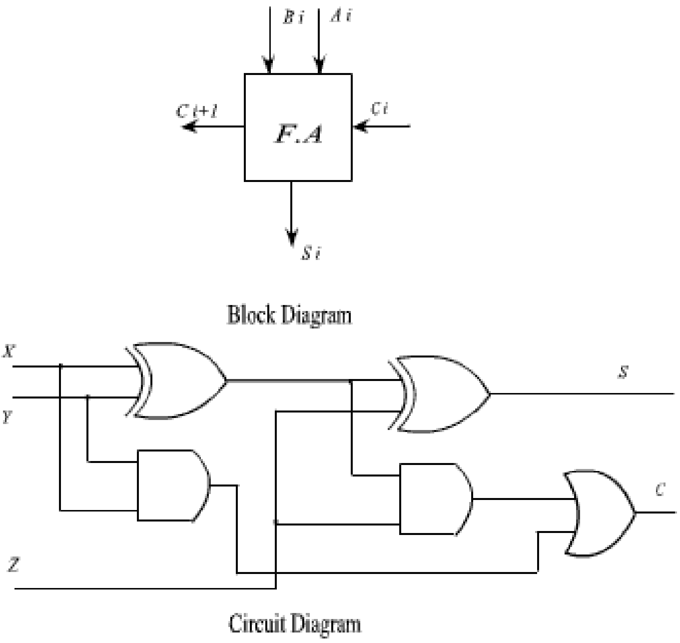


Figure: circuit and block diagram of full adder

4-bit full adder

Let's consider two 4 bit numbers

A =1 0 0 1 B = 0 0 1 1

The addition truth table is as follows

subscript	***i***	3	2	1	0
Input carry	C_i	0	1	1	0
Augends	A_i	1	0	0	1
Addend	B_i	0	0	1	1
Sum	S_i	1	1	0	0
Output carry	C_{i+1}	0	0	1	1

To get the four bit adder, we have to use 4 full adder blocks. The carry output the lower bit is used as a carry input to the next higher bit. The circuit of 4-bit adder is shown as given below:

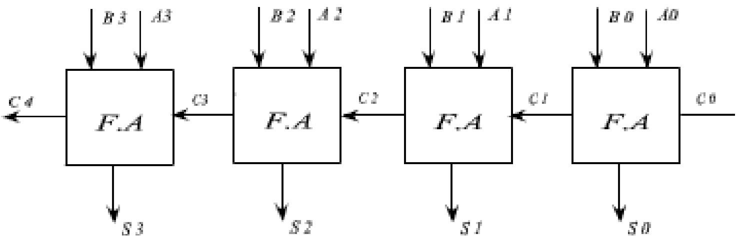


Figure: basic circuit diagram of a 4-bit full adder

Half Substractor

A combinational circuit which performs the subtraction of two bits is called half subtractor. The input variables designate the minuend and the subtrahend bit, whereas the output variables produce the difference and borrow bits.

TRUTH TABLE:

INPUT		OUTPUT	
A	**B**	**DIFF**	**BORR**
0	**0**	**0**	**0**
0	**1**	**1**	**1**
1	**0**	**1**	**0**
1	**1**	**0**	**0**

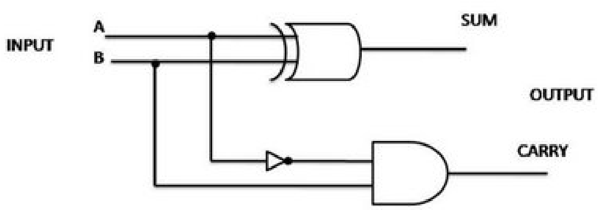
DESIGN:

From the truth table the expression for difference and borrow bits of the output can be obtained as,

Difference, DIFF = A $\oplus$ B

Borrow, BORR = A'B

CIRCUIT DIAGRAM:



Half Subtractor

FULL SUBTRACTOR

A combinational circuit which performs the subtraction of three input bits is called full subtractor. The three input bits include two significant bits and a previous borrow bit.

TRUTH TABLE:

S.No	INPUT			OUTPUT	
	A	B	C	DIFF	BORR
1.	0	0	0	0	0
2.	0	0	1	1	1
3.	0	1	0	1	1
4.	0	1	1	0	1
5.	1	0	0	1	0
6.	1	0	1	0	0
7.	1	1	0	0	0
8.	1	1	1	1	1

DESIGN:

From the truth table the expression for difference and borrow bits of the output can be obtained as,

Difference, DIFF= A'B'C + A'BC' + AB'C' + ABC

Borrow, BORR = A'B'C + A'BC' + A'BC+ABC

Using Karnaugh maps the reduced expression for the output bits can be obtained as,

DIFFERENCE

	B'C'(00)	B'C(01)	BC(11)	BC'(10)
A'(0)	0	1	0	1
A(1)	1	0	1	0

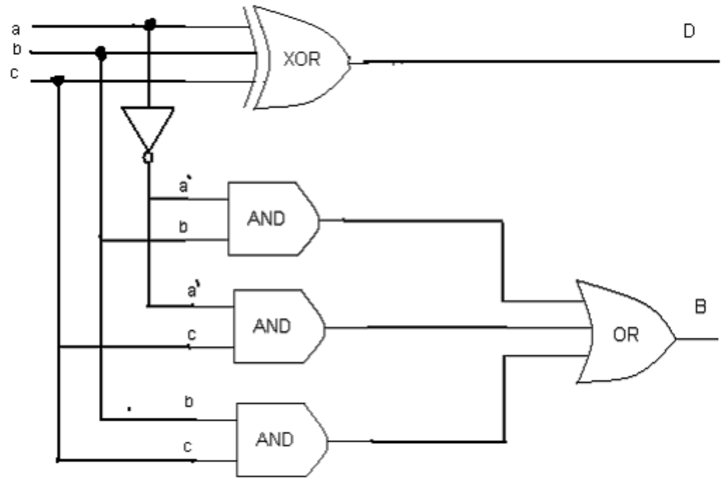
DIFF = A'B'C + A'BC' + AB'C' + ABC = A⊕B⊕C

BORROW

	B'C'(00)	B'C(01)	BC(11)	BC'(10)
A'(0)	0	1	1	1
A(1)	0	0	1	0

BORR = A'C+A'B+BC

CIRCUIT DIAGRAM:



BINARY SUBSTRACTOR FROM BINARY ADDER

The subtraction operation can be implemented with the help of binary adder circuit, because A-B=A+ (-B).

So we get as A-B

A-B=A+ (-B)

=A+ (2's complement of B)

=A+ (1's complement of B +1)

=A+ (Inverted form of B + 1)

i.e. if we place an inverter between B & full adder and set C0=1 then we get a subs tractor circuit from adder circuit.

To implement Adder & subtractor using same circuit we can place

1. XOR gate between B & Adder
2. Set C0=1
3. One mode selection line (M) which will determine the operation If, M=0 then A+B and if M=1 then A-B=A+ (-B) = A+1's complement of B+1.

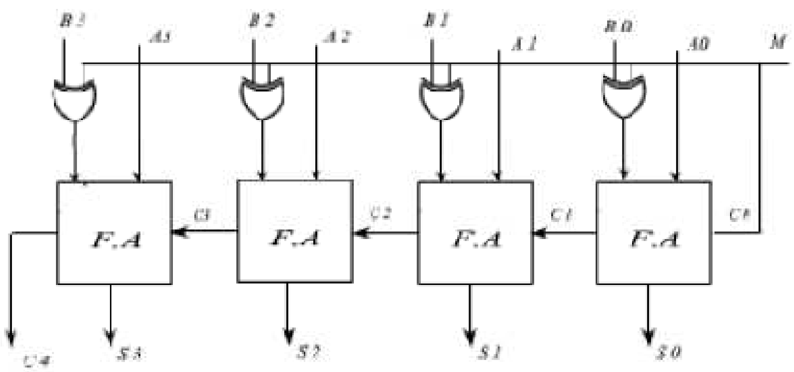


Figure: Figure: basic circuit diagram of a 4-bit adder-subtractor circuit