EXPT NO.: 3

DESIGN OF HALF AND FULL SUBTRACTOR

AIM:

To design and simulate half subtractor, full subtractor using basic gates and full subtractor using 2 half subtractors.

HALF SUBTRACTOR USING BASIC GATES:

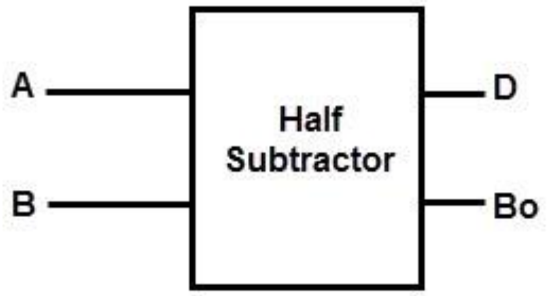
A half subtractor is a multiple output combinational logic network that does the subtraction of two bits of binary data. It has input variables and two output variables. Two inputs are corresponding to two input bits and two output variables corresponds to the difference bit and borrow bit.

The binary subtraction is also performed by the Ex-OR gate with additional circuitry to perform the borrow operation. Thus, a half subtractor is designed by an Ex-OR gate including AND gate with the A input complemented before feeding into the gate.

$$\textbf{Difference (D)} = A \oplus B$$

$$\textbf{Borrow } (B_0) = \bar{A}.B$$

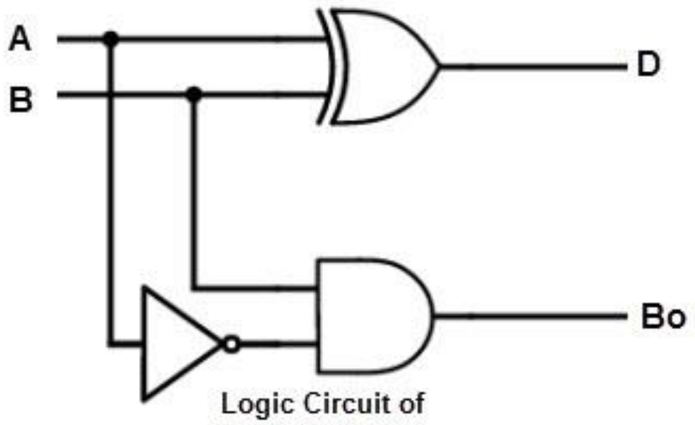
The block diagram, logic diagram and the truth table for half adder is given below:



Block Diagram

A	B	D	B_o
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Truth Table



Logic Circuit of Half Subtractor

The i/o ports needed to be declared for the formation of half subtractor is given below:

Port Name	INPUT/OUTPUT	Bus
A	In	No
B	In	No
D	Out	No
Bout	Out	No

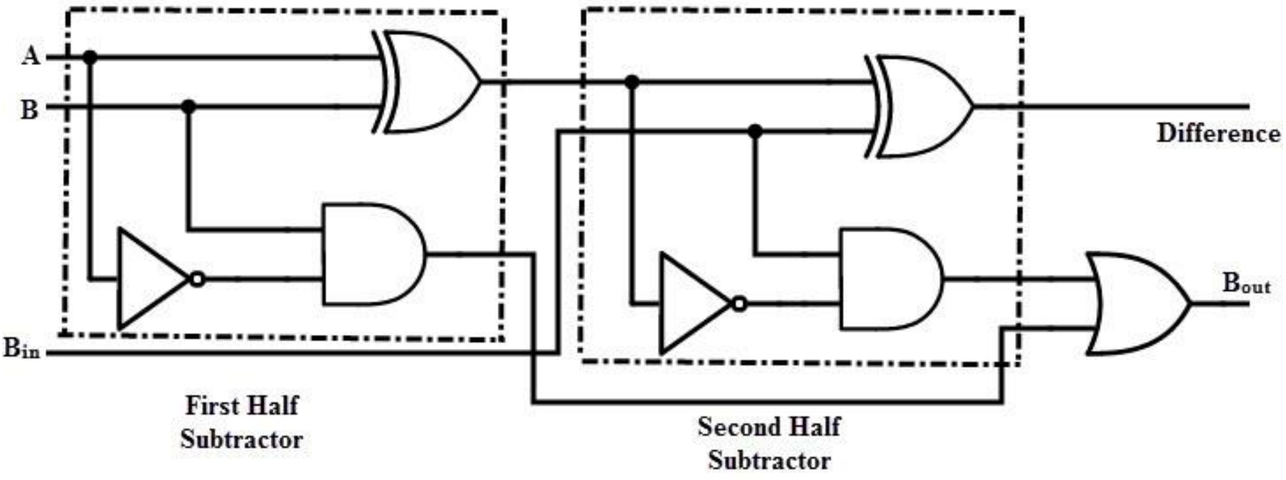
NB: Use temporary variable where ever necessary.

FULL SUBTRACTOR USING BASIC GATES:

A combinational logic circuit performs a subtraction between the two binary bits by considering borrow of the lower significant stage is called as the full subtractor. In this, subtraction of the two digits is performed by taking into consideration whether a 1 has already been borrowed by the previous adjacent lower minuend bit or not.

It has three input terminals in which two terminals corresponds to the two bits to be subtracted (minuend A and subtrahend B), and a borrow bit B_i corresponds to the borrow operation. There are two outputs, one corresponds to the difference D output and other borrow output B_o as shown in figure along with truth table.

The circuit diagram and truth table of full subtractor is given below:



TRUTH TABLE:

A	B	C	DIFFERENCE	BORROW
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

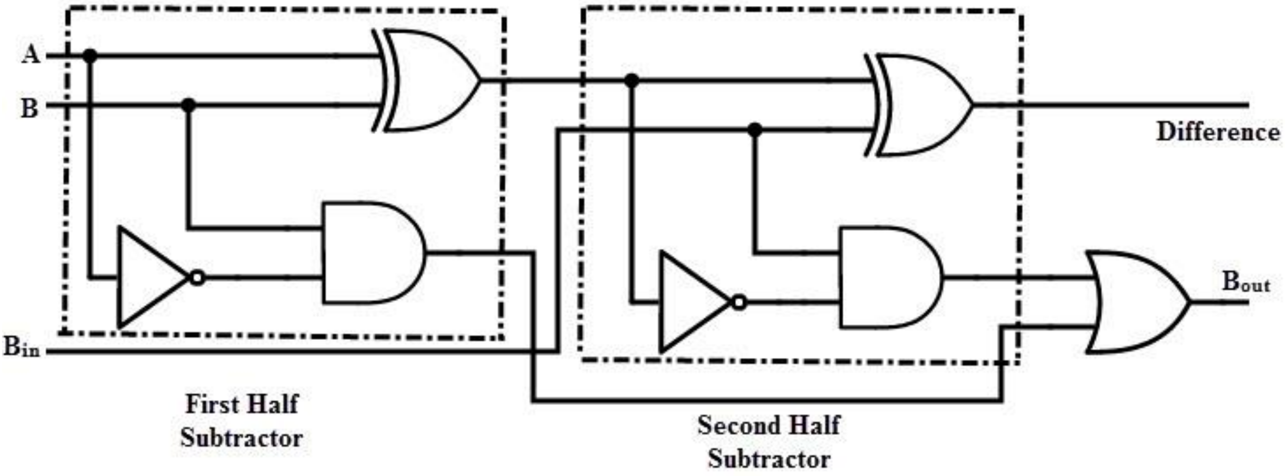
The i/o ports needed to be declared for the formation of full subtractor is given below:

Port Name	INPUT/OUTPUT	Bus
A	In	No
B	In	No
Bin	In	No
Diff	Out	No
Bout	Out	No

NB: Use temporary variable where ever necessary.

FULL SUBTRACTOR USING HALF SUBTRACTORS AS COMPONENTS:

The block diagram of a full subtractor constructed using half subtractor is given below:



The i/o ports needed to be declared for the formation of full subtractor is given below:

Port Name	INPUT/OUTPUT	Bus
A	In	No
B	In	No
Bin	In	No
Diff	Out	No
Bout	Out	No

NB: Use temporary variable where ever necessary.

The vhdl code for a full subtractor using 2 half subtractors will be:

```
entity fullsub is
   Port ( A : in  STD_LOGIC;
          B : in  STD_LOGIC;
          Bin : in  STD_LOGIC;
          Diff : out  STD_LOGIC;
          Borr : out  STD_LOGIC);
end fullsub;

architecture Behavioral of fullsub is
component halfsub is
   Port ( a : in  STD_LOGIC;
          b : in  STD_LOGIC;
          D : out  STD_LOGIC;
          Bor : out  STD_LOGIC);
end component;
 signal temp, c1, c2 : STD_LOGIC := '0';

begin
HS0: halfsub port map(a=>A, b=>B, D=>temp, Bor=>c1);
HS1: halfsub port map(a=>temp, b=>Bin, D=>Diff, Bor=>c2);
Borr <= c1 or c2;

end Behavioral;
```