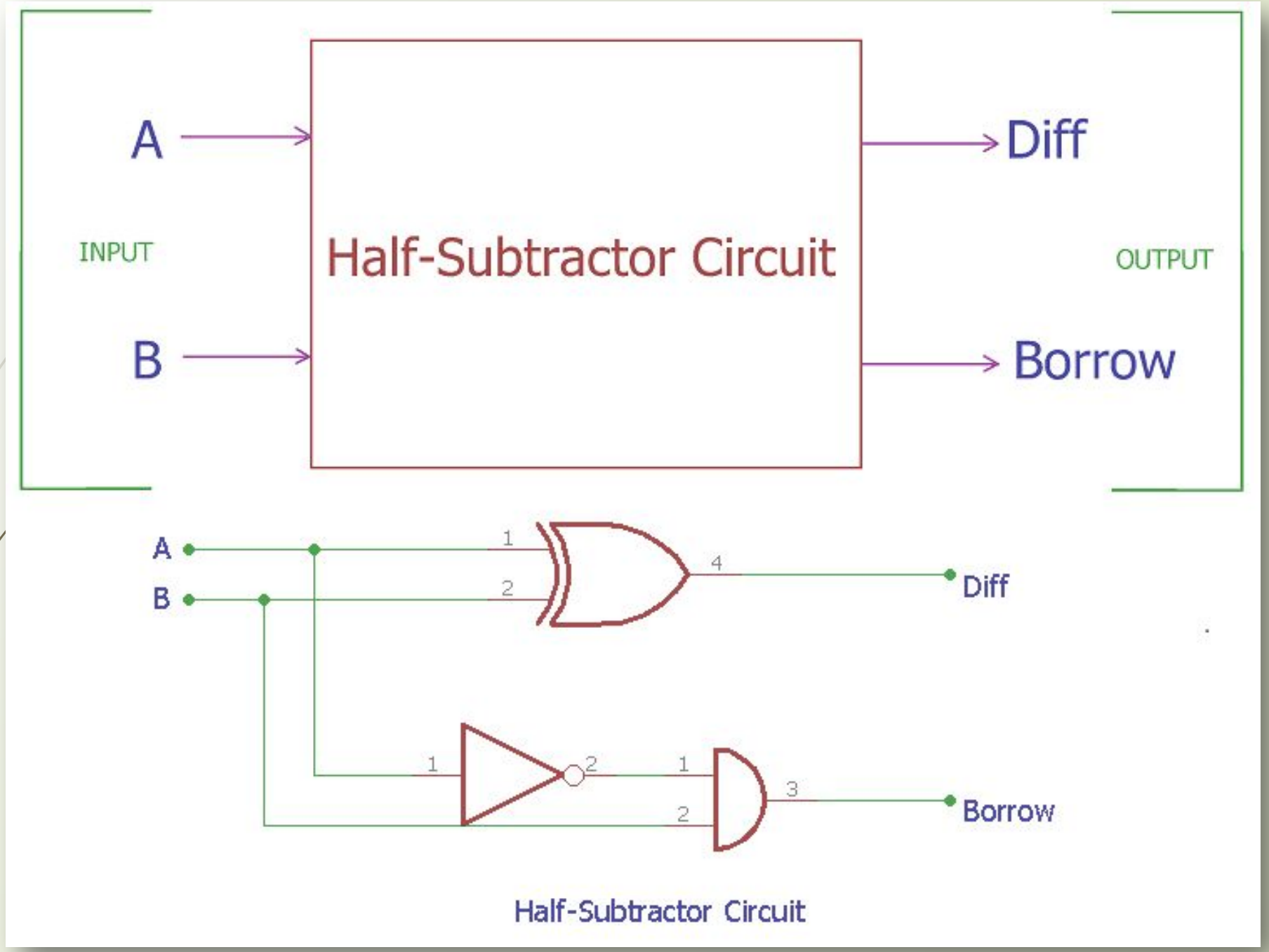
VHDL CODE FOR HALF AND FULL SUBTRACTOR


A
B
INPUT
Half-Subtractor Circuit
Diff
Borrow
OUTPUT
A
B
1
2
4
Diff
1
2
1
2
3
Borrow
Half-Subtractor Circuit

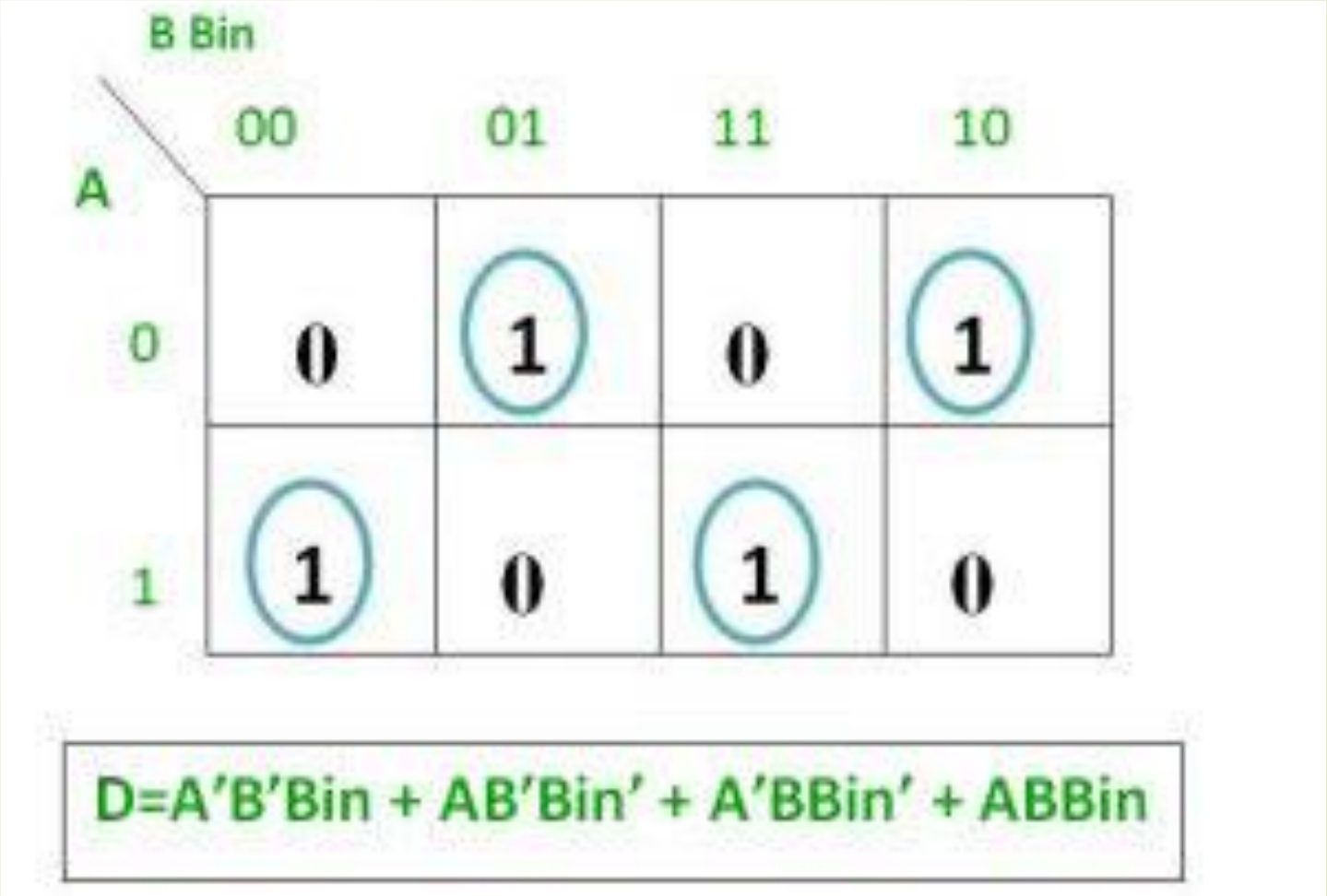
1st Bit or Digit	*2nd Bit or Digit*	*Difference*	*Borrow*
0	0	0	0
1	0	1	0
0	1	1	1
1	1	0	0

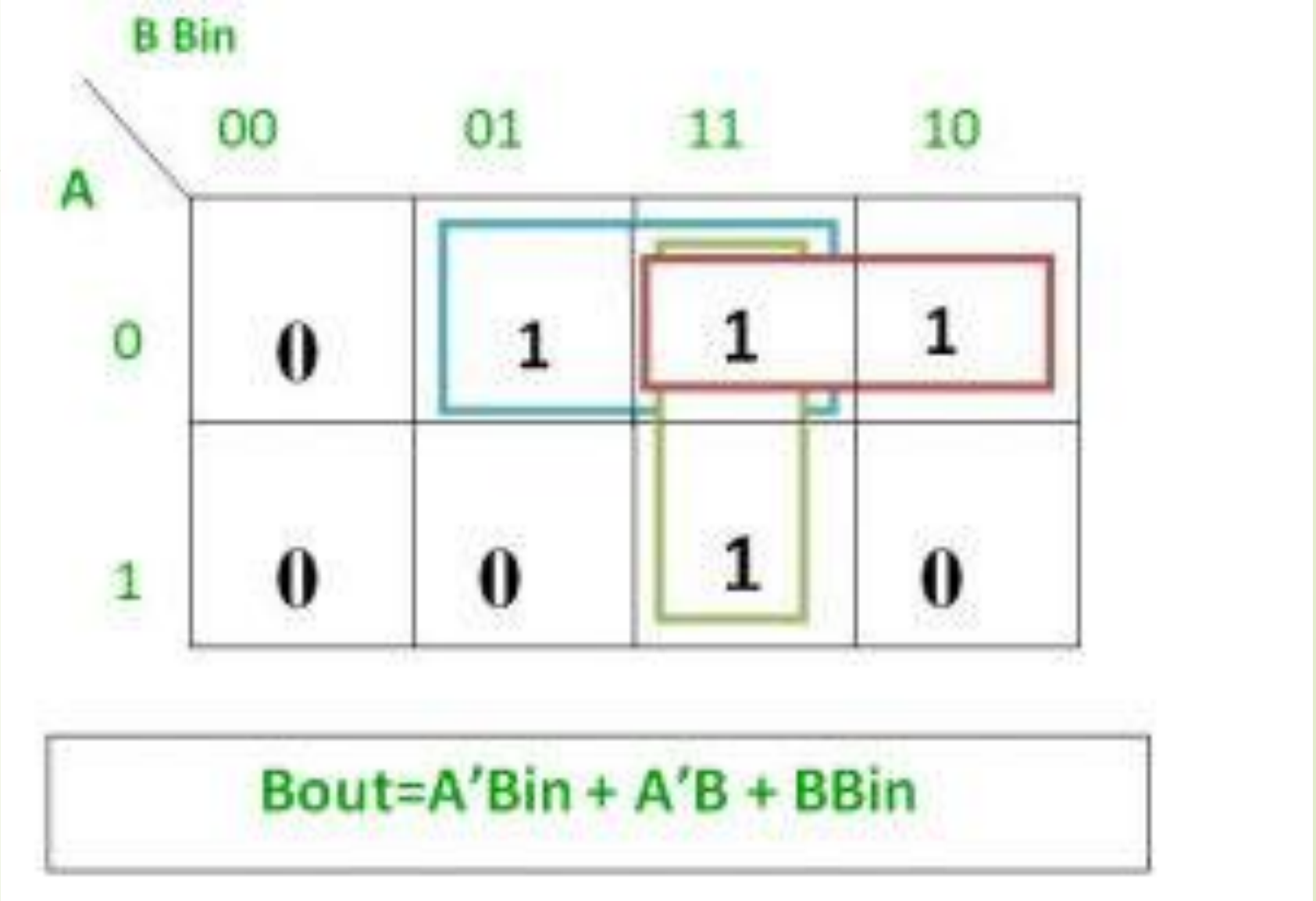
VHDL code for Half Subtractor:

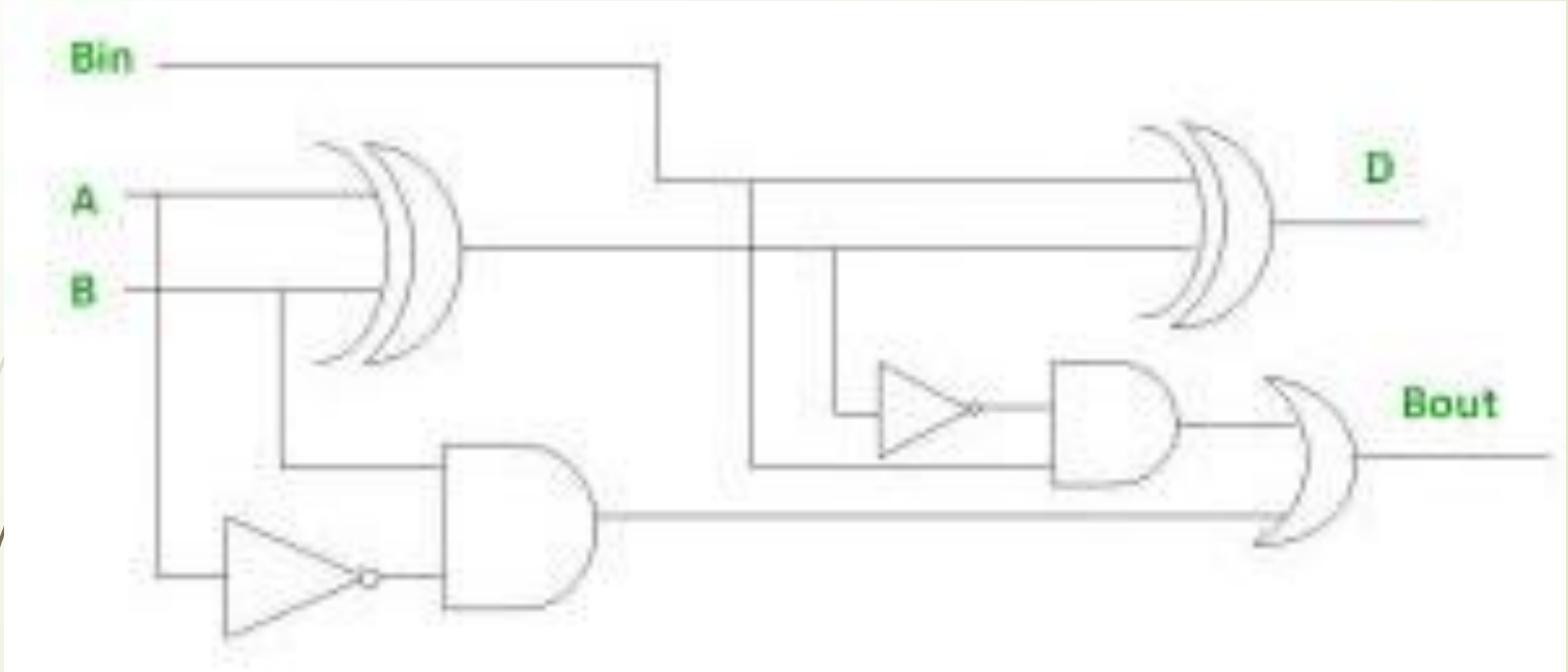
```
library IEEE;
use IEEE.std_logic_1164.all;
entity half_sub is
port(A,B: in std_logic;
diff,borrow : out std_logic
);
end half_sub;
architecture flow of half_sub is
begin
diff<= A xor B;
borrow<=not(A) and B;
end flow;
```

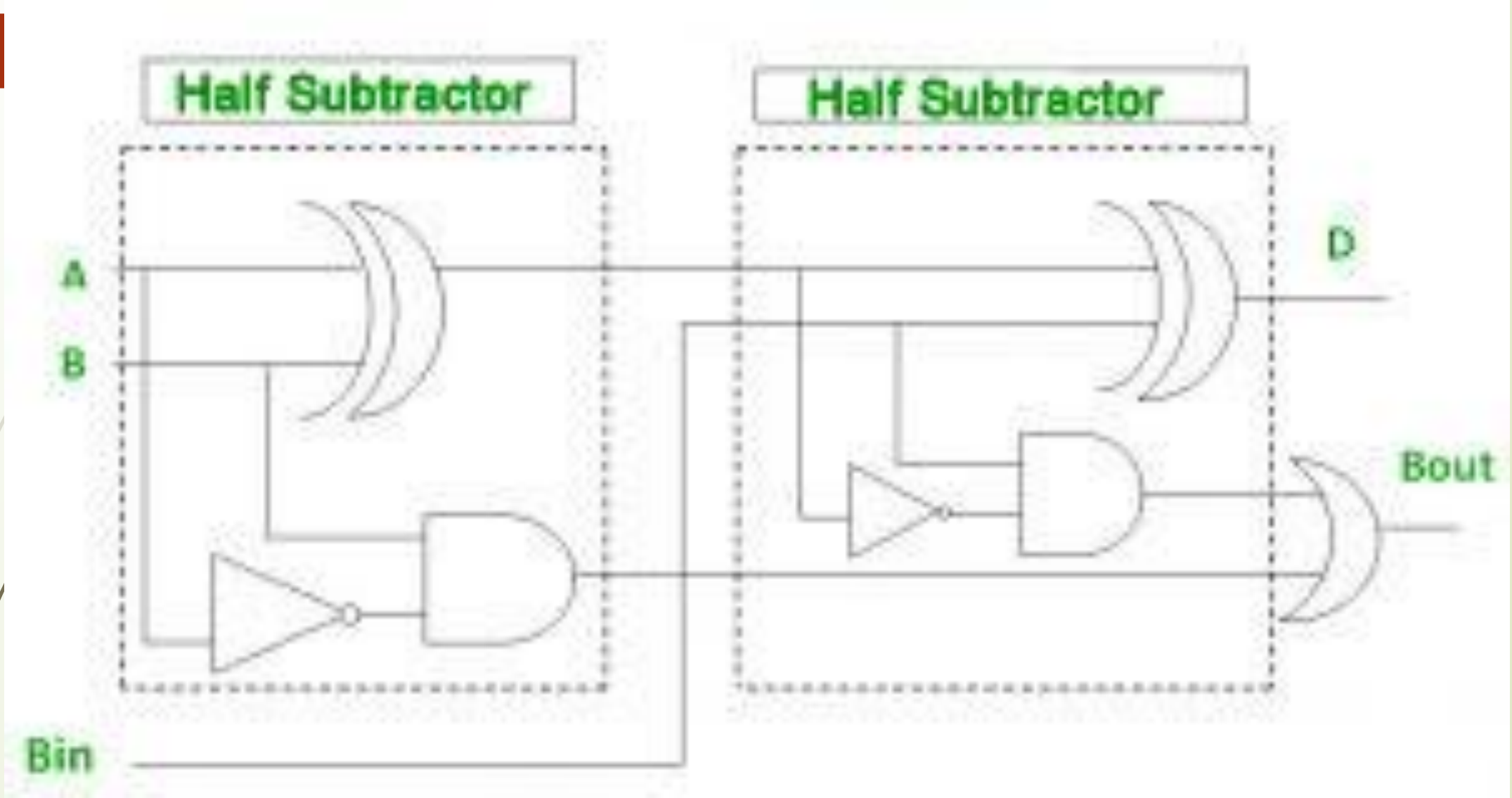
TRUTH TABLE OF FULL SUBTRACTOR

INPUT			OUTPUT	
A	B	Bin	D	Bout
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

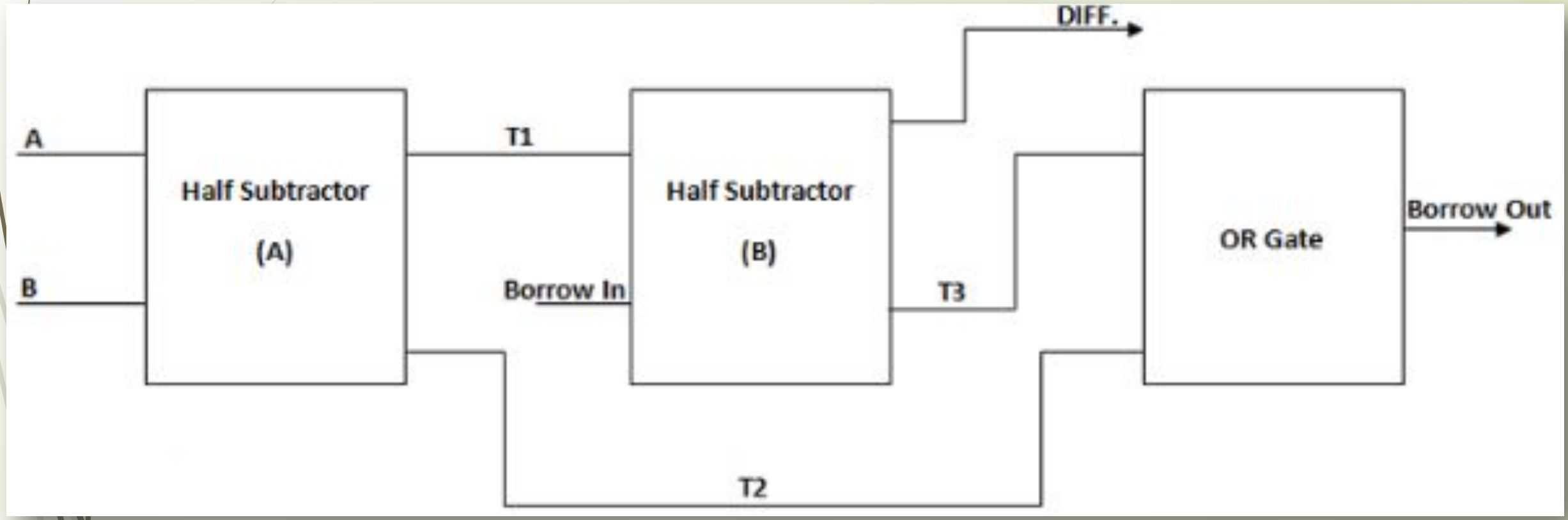

B Bin
00
01
11
10
A
0
1
0
1
0
1
1
0
1
0
D=A'B'Bin + AB'Bin' + A'BBin' + ABBin


B Bin
00
01
11
10
A
0
1
0
1
1
1
0
0
1
0
Bout=A'Bin + A'B + BBin


Bin
A
B
D
Bout


Half Subtractor
Half Subtractor
A
B
D
Bout
Bin

Full Subtractor



```
library IEEE;
use IEEE.std_logic_1164.all;
entity full_sub is
    port(A,B,borrow_in: in std_logic;
         diff,borrow_out : out std_logic
);
end full_sub;
architecture flow of full_sub is
component half_sub is
    port(A,B : in std_logic;
         diff,borrow : out std_logic
);
end component;
component or_gate is
    port(A,B : in std_logic;
         Y : out std_logic
);
end component ;
signal T1,T2,T3:std_logic;
begin
half_sub1: half_sub port map (A=>A,B=>B,diff=>T1,borrow=>T2);
half_sub2: half_sub port map(A=>T1,B=>borrow_in,diff=>diff,borrow=>T3);
or_gate1: or_gate port map(A=>T3,B=>T2,Y=>borrow_out);
end flow;
```