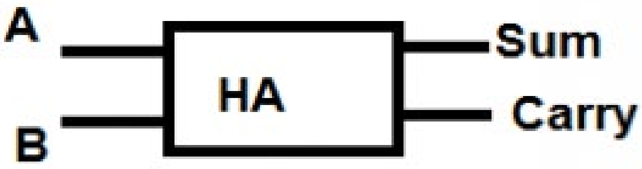
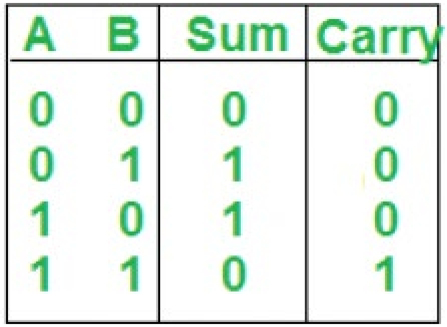
Half Adder

Half Adder- The addition of 2 bits is done using a combination circuit called a Half adder. The input variables areaugend and addend bits and output variables are sum & carry bits. A and B are the two input bits.



Half Adder

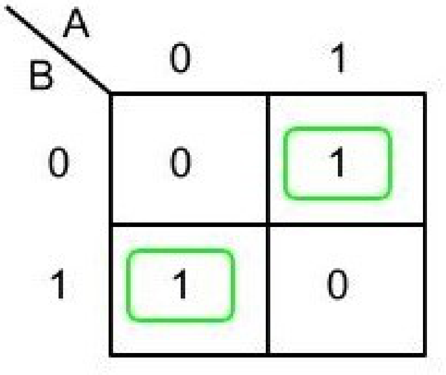
Truth Table -



A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Logical Expression:

1) **For Sum:**



B \ A	0	1
0	0	1
1	1	0

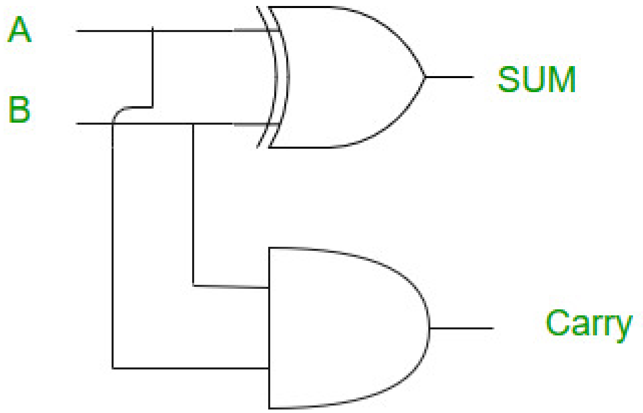
Sum =A X-OR B

2) For Carry:

B \ A	0	1
0	0	0
1	0	1

$$Carry = A \cdot B$$

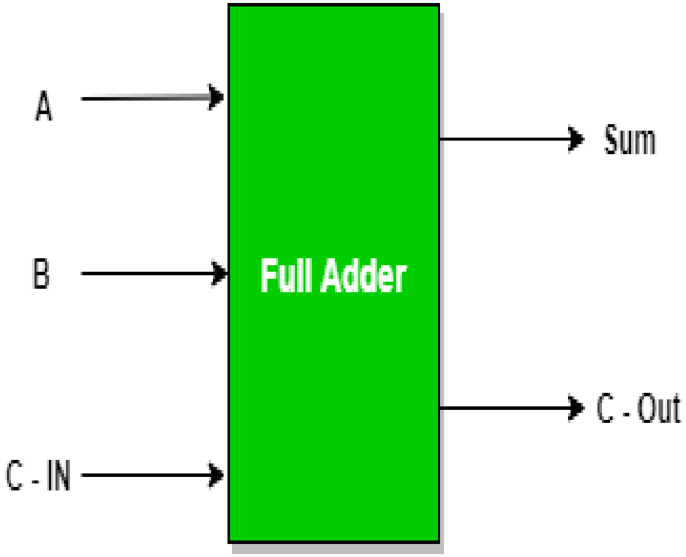
Implementation:



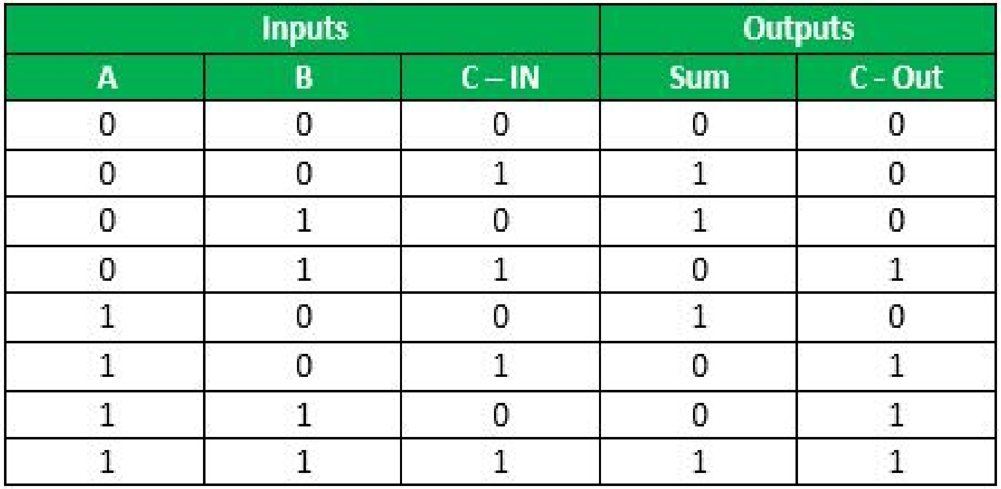
Note: Half adder has only two inputs and there is no provision to add a carry coming from the lower order bits when multi addition is performed.

Full Adder

Full Adder : It is the adder which adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM.
A full adder logic is designed in such a manner that can take eight inputs together to create a byte-wide adder and cascade the carry bit from one adder to the another.



Truth Table:



Inputs			Outputs	
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

Logical Expression for SUM:

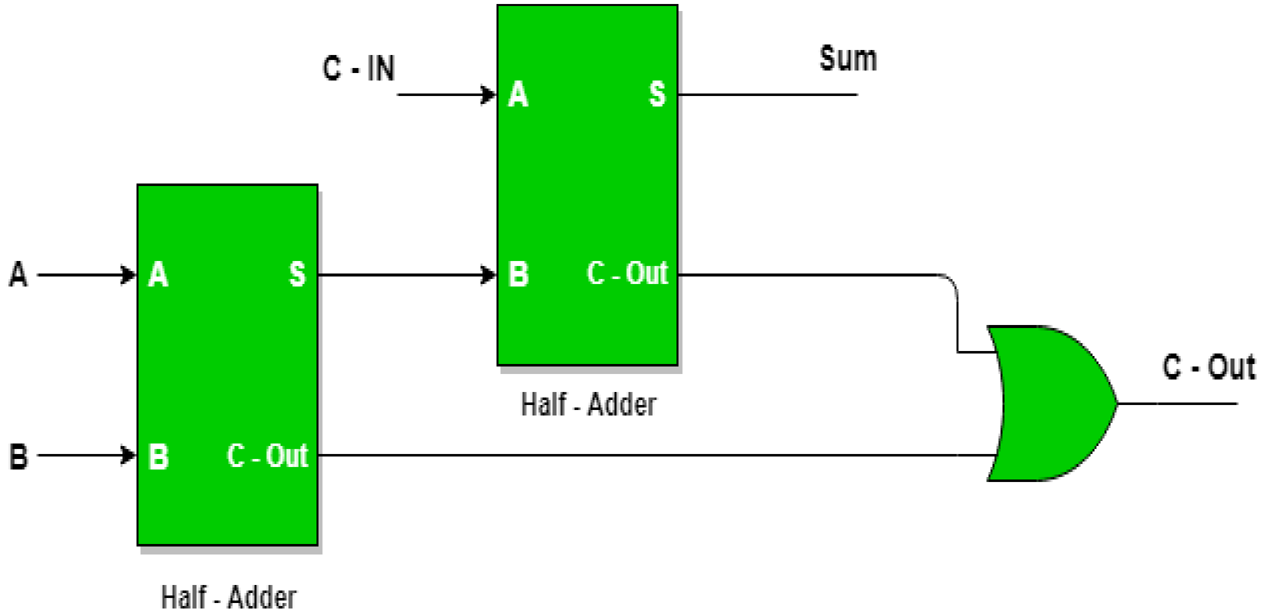
= A’ B’ C-IN + A’ B C-IN’ + A B’ C-IN’ + A B C-IN
= C-IN (A’ B’ + A B) + C-IN’ (A’ B + A B’)
= C-IN X-OR A X-OR B
= $\sum$m (1,2,4,7)

Logical Expression for C-OUT:

= A B + B C-IN + A C-IN
= $\sum$m (3,5,6,7)

Implementation of Full Adder using Half Adders

2 Half Adders and a OR gate is required to implement a Full Adder.



With this logic circuit, two bits can be added together, taking a carry from the next lower order of magnitude, and sending a carry to the next higher order of magnitude.

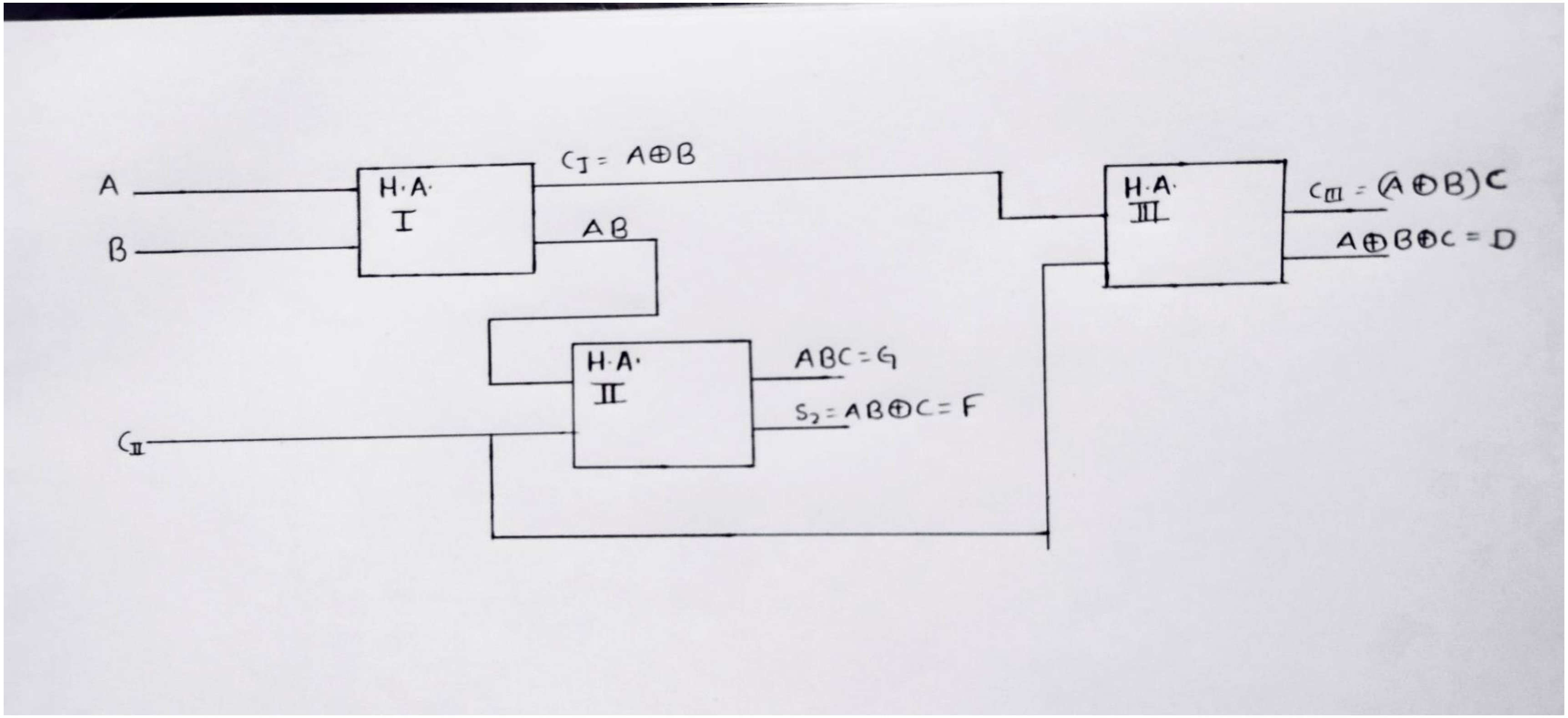
Q1. Design and implement the following using half adder only.

D= A XOR B XOR C

E= A’BC+AB’C =(A’B+AB’)C =(A XOR B)C

F= AB’C+(A’+B’)C = ABC’+A’B’C

G= ABC



BCD ADDER

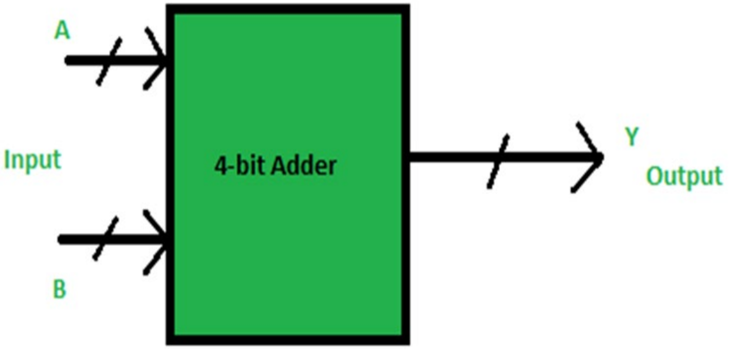
Procedure for BCD addition

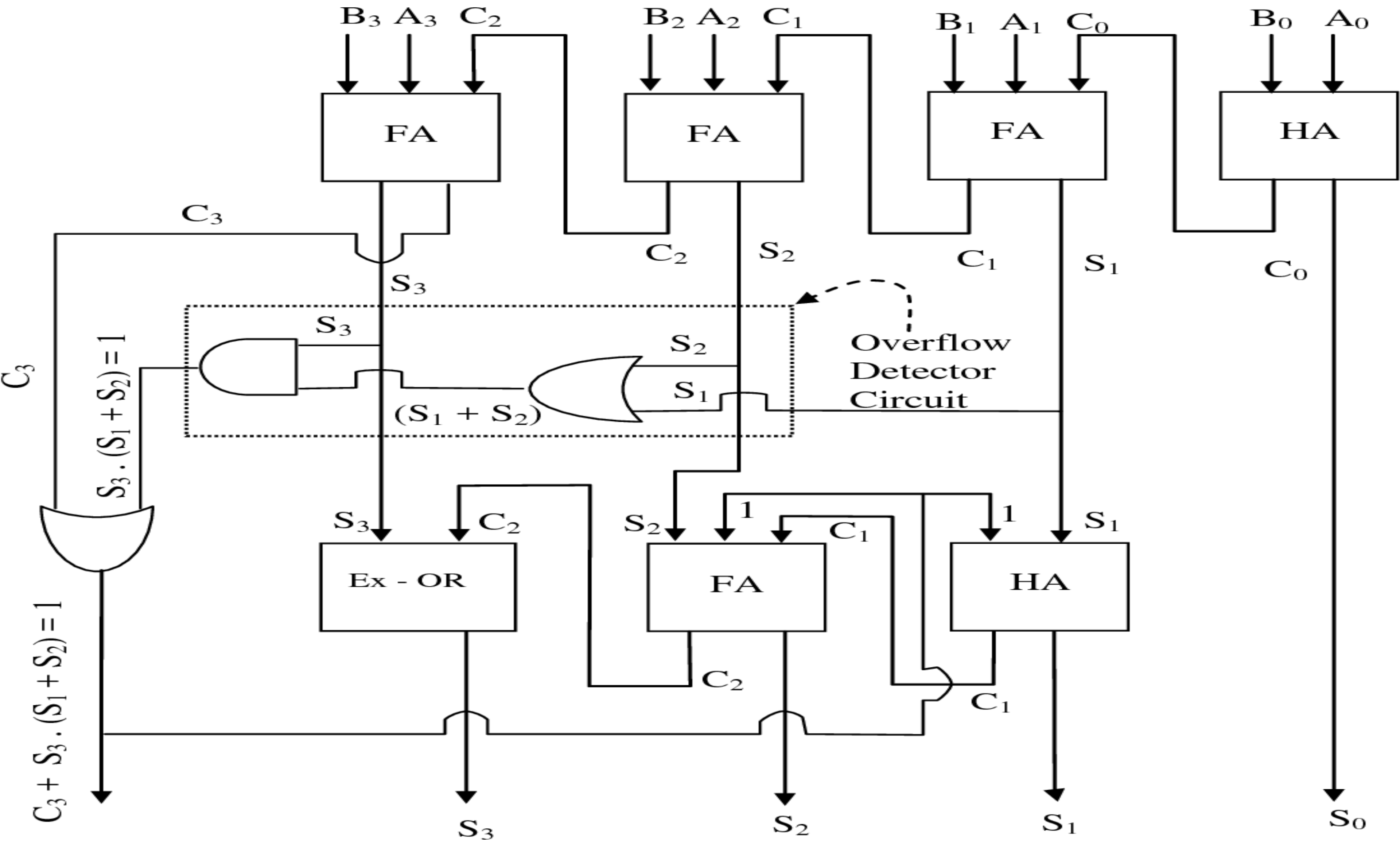
1. Add two BCD numbers using ordinay binary addition.
2. If four-bit sum is equal to or less than 9, no correction is needed. The sum is in proper BCD form.
3. If the four-bit sum is greater than 9 or if a carry is generated from the four-bit sum, the sum is invalid.
4. To correct the invalid sum, add 0110_2 to the four-bit sum. If a carry results from this addition, add it to the next higher-order BCD digit.

Let A_3 , A_2 , A_1 , A_0 and B_3 , B_2 , B_1 , B_0 are two four – bit BCD numbers. these two BCD no. are added along with the carry in using 4- bit parallel binary adder as given below

1st no.	A_3	A_2	A_1	A_0
2nd no.	B_3	B_2	B_1	B_0
	S_3	S_2	S_1	S_0
	C_3	C_2	C_1	C_0

S_3	S_2	S_1	S_0
0	Y	Y	0
0	0	0	0
0	1	1	0





BCD Adder Block Diagram

The result obtained will be a binary no i.e; (C3 S3 S2 S1 S0) The logical circuit AND / OR gate is used to check the binary result obtained .

If (S3 S2 S1 S1) greater than 9 than , A=1 or B=1 , Hence Y=1.

similarly if last carry C3 , then Y=1.

So when Y=1 , 6 (0 1 1 0) is added with a binary result.

If Y= 0 then , 0 (0 0 0 0) is added with a binary result .

Using half adder zero (HA_0) full adder (FA_4) and (HA_1) the addition of two numbers is performed as given below-

carry generated	C2’	C1’			
1st No.	S3	S2	S1	S0	
2ND NO.	0	Y	Y	0	
	Z3	Z2	Z1	Z0	--------→ BCD RESULT