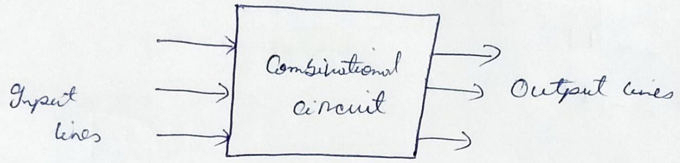
Ch-4 - Combinational logic

Combinational circuit a type of digital logic circuit whose outputs depends on the present input values only and does not depend on past output and input values.

- It doesnot have any feedback path between input and output.
- Open loop system.



- Three key elements.

- Input lines - Enter input values into the Combinational circuit.
- Processing Unit - Main element.
  - Processes the input values depending on the types of circuit.
- Output lines - Take results generated by the circuit.

- Characteristic of CC -

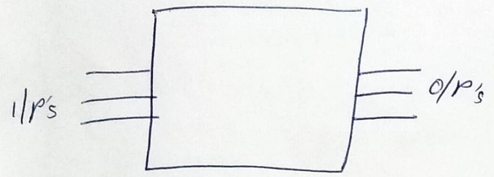
- The output of a CC at any instance of time, depends only on the present input values at that instant of time.
- CC do not use any kind of memory element in their circuits. Thus they do not store the previous state of input and output.
- The output of CC can entirely predicted using its logical operation and input values.
- CC produce an instantaneous output in response to any change in its input values.

Types-

(i) Binary Adder (HA, FA) (ii) Binary Substraction (HS, FS) (iii) Multiplexers (iv) Demux

(v) Encoders (vi) Decoders (vii) Comparators

Designing of Combinational circuit :



Step 1- Determine and define total I/p's and total O/p's of the circuit.

Step 2- Make truth table that defines relationship in betn I/p's & O/p's

Step 3. Determine boolean eqn using K-map.

Step 4. Based on BE, we can form circuit.

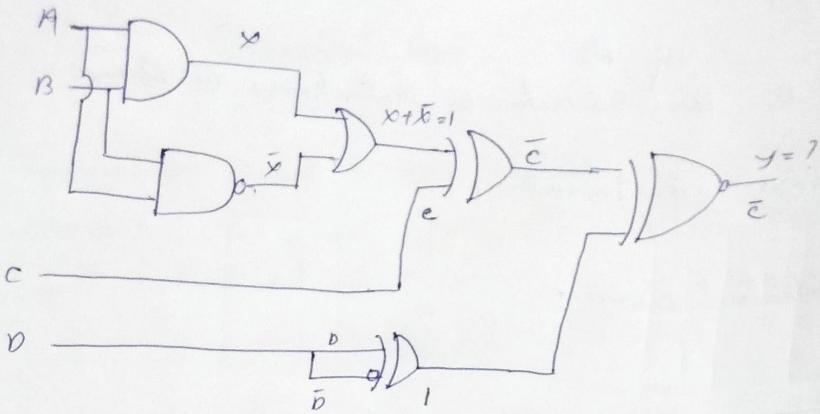
Q:- The minimal function that can detect "divisible by 2' with 8421 BCD $[D_8 D_4 D_2 D_1]$ is given by $\bar{D}_1$

⇒

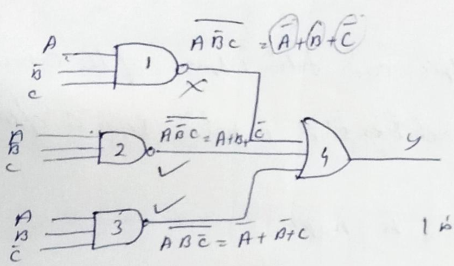
	D_8	D_4	D_2	D_1	Y
0 -	0	0	0	0	1
1 -	0	0	0	1	0
2 -	0	0	1	0	1
3 -	0	0	1	1	0
4 -	0	1	0	0	1
5 -	0	1	0	1	0
6 -	0	1	1	0	1
7 -	0	1	1	1	0
8 -	1	0	0	0	1
9 -	1	0	0	1	0
10 -	1	0	X	0	X
⋮					⋮
15 -	1	1	X	1	X

$D_2 D_1$ \ $D_8 D_4$	00	01	11	10
00	1_0	1_4	X_{12}	1_8
01	1	5	13	9
11	3	7	15	11
10	1_2	1_6	X_{14}	X_{10}

$\bar{D}_1$



②

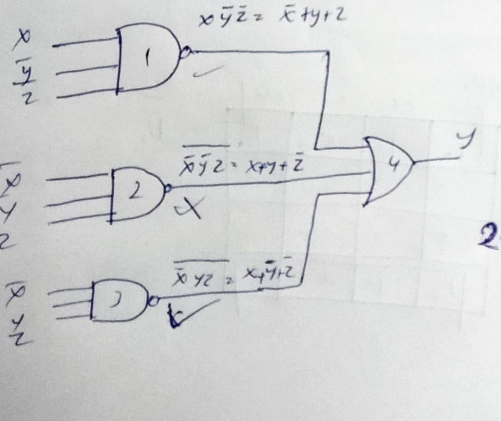


What is redundant gate in given CC?

1 is redundant.

Check with other gate.. If its present in other gate then it is redundant gate.

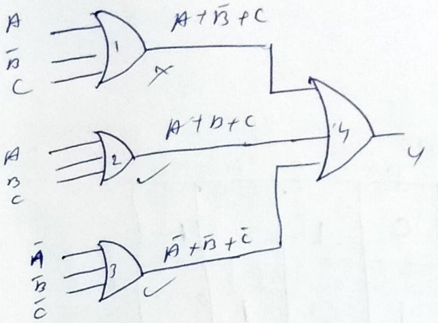
③



Find redundant gate?

2 is redundant.

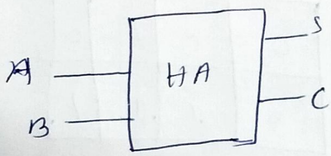
(iv)



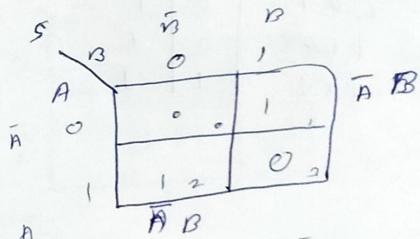
Redundant gate — 1

Half Adder :

⇒ H A is a ckt where we perform 2-bit addn

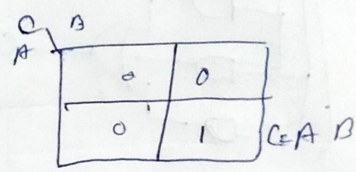


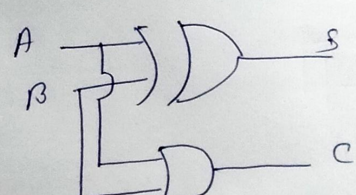
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



$S = A\bar{B} + \bar{A}B$

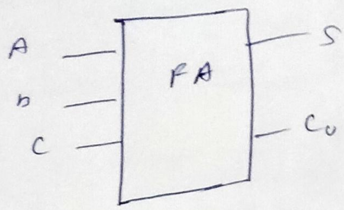
$= A \oplus B$





Full Adder

= Used to perform 3 bit addⁿ.



S: A \ BC	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$S = A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC + \bar{A}B\bar{C}$

$= \bar{A}(\bar{B}C + B\bar{C}) + A(BC + \bar{B}\bar{C})$

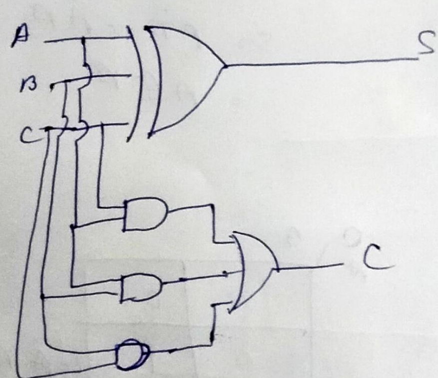
$= \bar{A}(B \oplus C) + A(B \odot C)$

$= \bar{A}(B \oplus C) + A(\overline{B \oplus C}) = A \oplus B \oplus C$

A	B	C	S	C_0
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

C: A \ BC	00	01	11	10
0	0	0	1	0
1	0	1	1	1

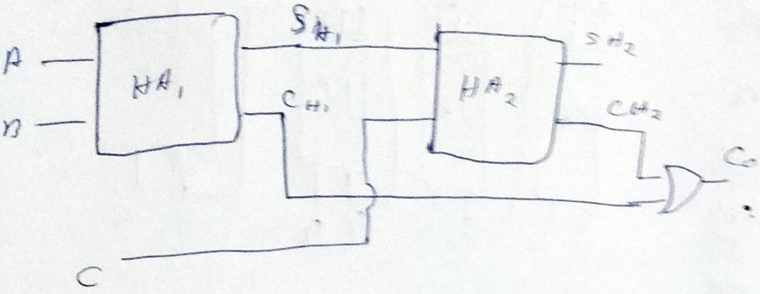
$\boxed{C = BC + AC + AB}$



Full Adder using Half Adder.

$A \oplus B$

$A\bar{B} + \bar{A}B$



$S_{H_1} = A \oplus B = A\bar{B} + \bar{A}B$

$C_{H_1} = AB$

$S_{H_2} = S_{H_1} \oplus C = A \oplus B \oplus C$

$C_{H_2} = S_{H_1} \cdot C = (A\bar{B} + \bar{A}B)C$

$= A\bar{B}C + \bar{A}BC$

$C_o = C_{H_2} + C_{H_1}$

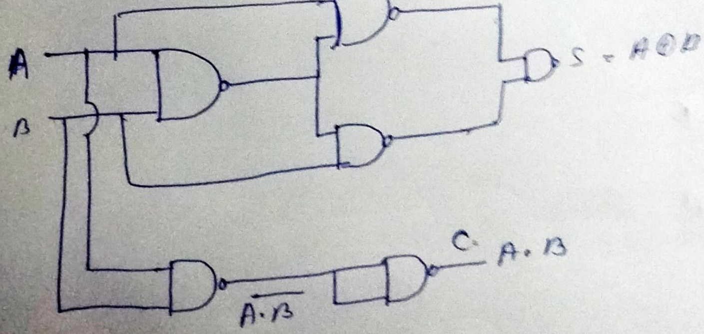
$= AB + A\bar{B}C + \bar{A}BC$

$= BC + AC + AB$

C

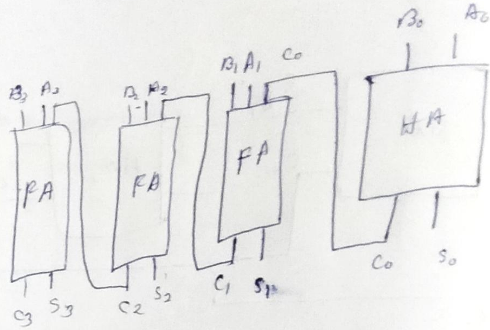
A \ BC	00	01	11	10
0			1	
1		1	1	1

Half Adder by NAND gates



4 Bits Parallel Adder.

		C_2	C_1	C_0	
A		A_3	A_2	A_1	A_0
B	+	B_3	B_2	B_1	B_0
C_0 S	C_3	S_3	S_2	S_1	S_0
		3	3	3	2bit
		FA			HA



Half Subtraction:

Two bits.

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

D

A \ B		
	0	1
	1	0

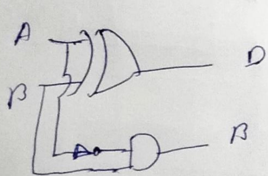
$D \quad A\bar{B} + \bar{A}B$

$D = A \oplus B$

B

A \ B		
	0	1
	0	0

$B = \bar{A}B$



Full Substraction

Three bit Substraction.

A	B	C	D	B_0
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

D

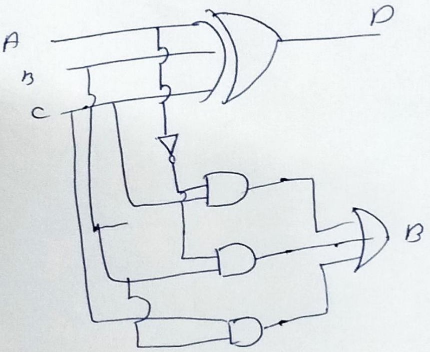
C \ AB	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$D = A \oplus B \oplus C$

B

C \ AB				
	0	1	0	0
	1	1	1	0

$C = \bar{A}C + \bar{A}B + BC$



Even Parity / Odd Parity Generator.

Even Parity generator parity = 1, when number of 1's at inputs are ODD.

'' '' '' '' Even.

'' '' = 1, '' '' ''

Odd '' '' ''

b_3	b_2	b_1	b_0	O/P (Even)
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

$b_1 b_0$ \ $b_3 b_2$	00	01	10	11
00	0	1		1
01	1	0	1	
10	0	1		1
11	1		1	

$P_{even} = b_0 \oplus b_1 \oplus b_2 \oplus b_3$

$O_{odd} = \overline{b_3 \oplus b_2 \oplus b_1 \oplus b_0}$