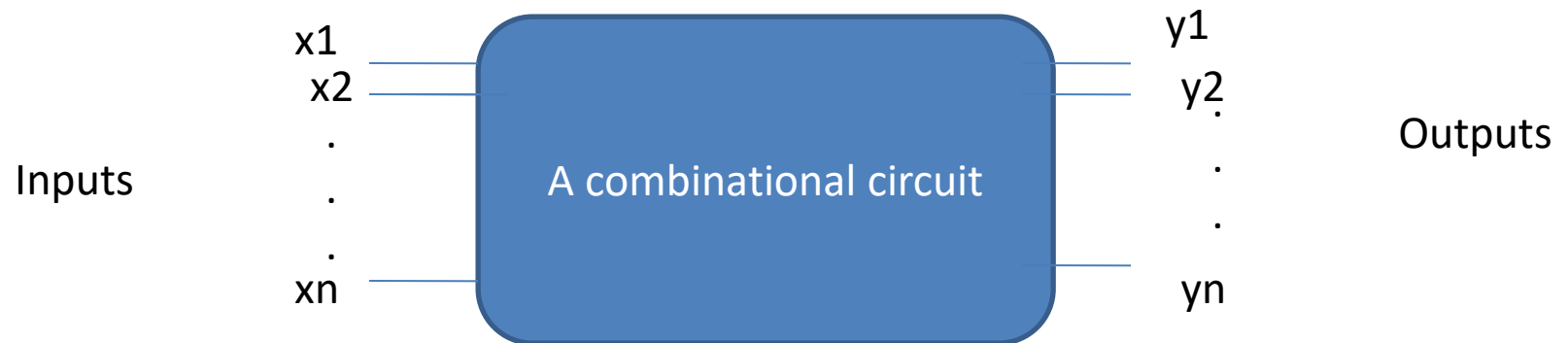


Introduction:

- Logic circuit for digital system may be:
 - Combinational Logic
 - Sequential Logic

Combinational logic

- Output at any times depends only on present combination of input.
- Combinational circuit does not need any memory.
- Combinational circuits are faster in operation than sequential circuits.



Block diagram

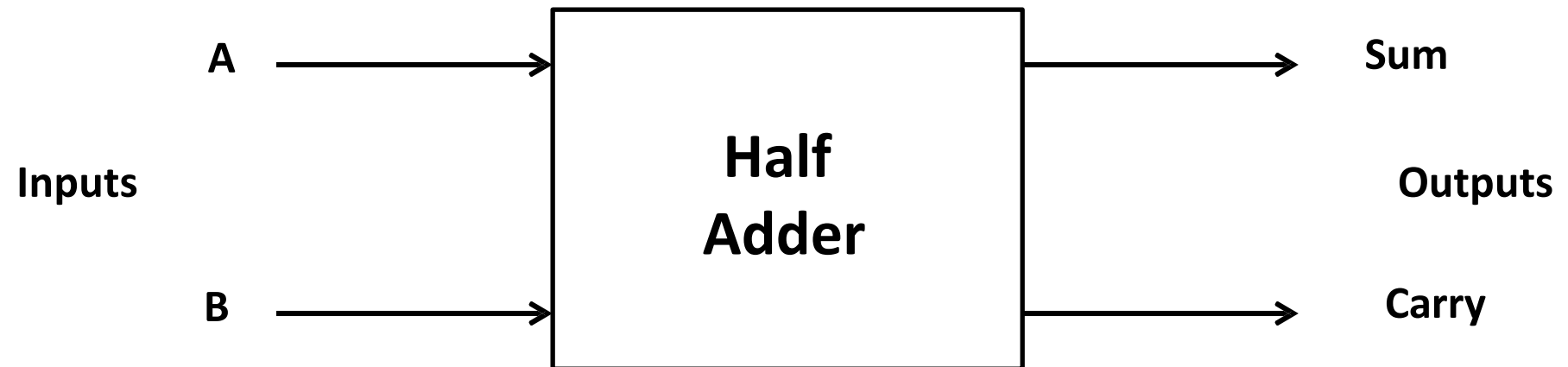
Design procedure:

The design steps for combinational logic are:

- Question is given to you.
- Find the number of i/p and o/p.
- Draw the truth table.
- Simplify the o/p using K-map. Simplification is done for o/p only.
- Draw the logic diagram.

Half Adder

- ✓ Half adder is a combinational logic circuit with two inputs and two outputs.
- ✓ It is a basic building block for addition of two single bit numbers.



Half Adder

Truth Table for Half Adder

Input		Output	
A	B	Sum (S)	Carry (C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Half Adder

K-map for Sum Output:

		B	
		\overline{B}	B
A	\overline{A} 0	0	1
	A 1	1	0

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

$$S = A \oplus B$$

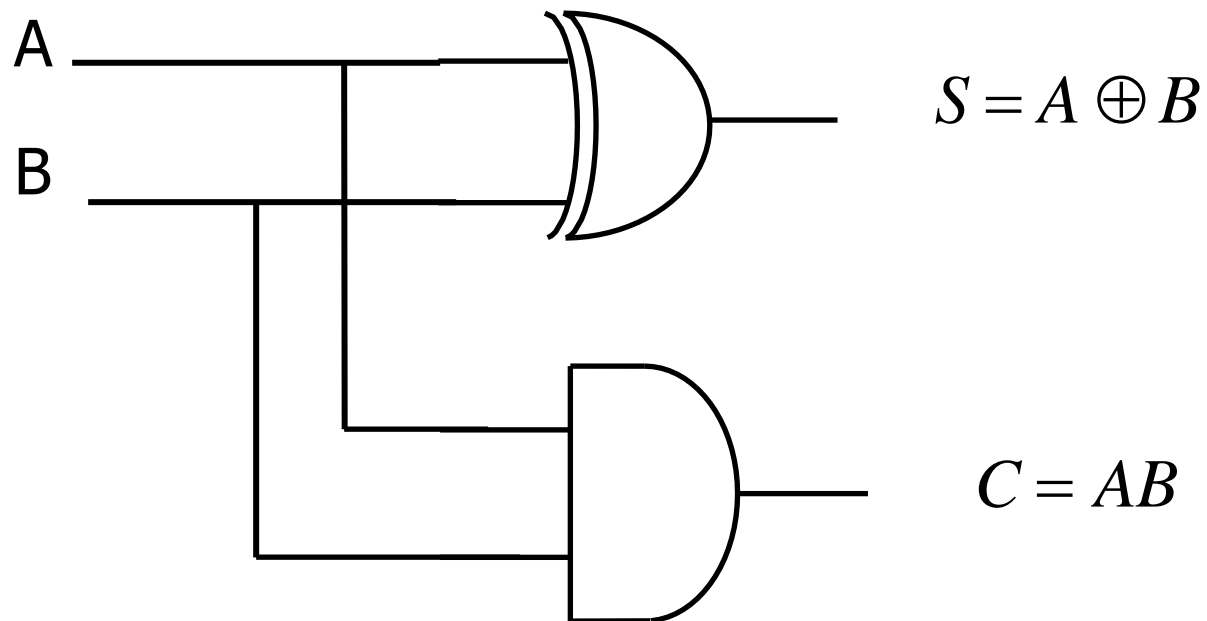
K-map for Carry Output:

		B	
		\overline{B}	B
A	\overline{A} 0	0	0
	A 1	0	1

$$C = AB$$

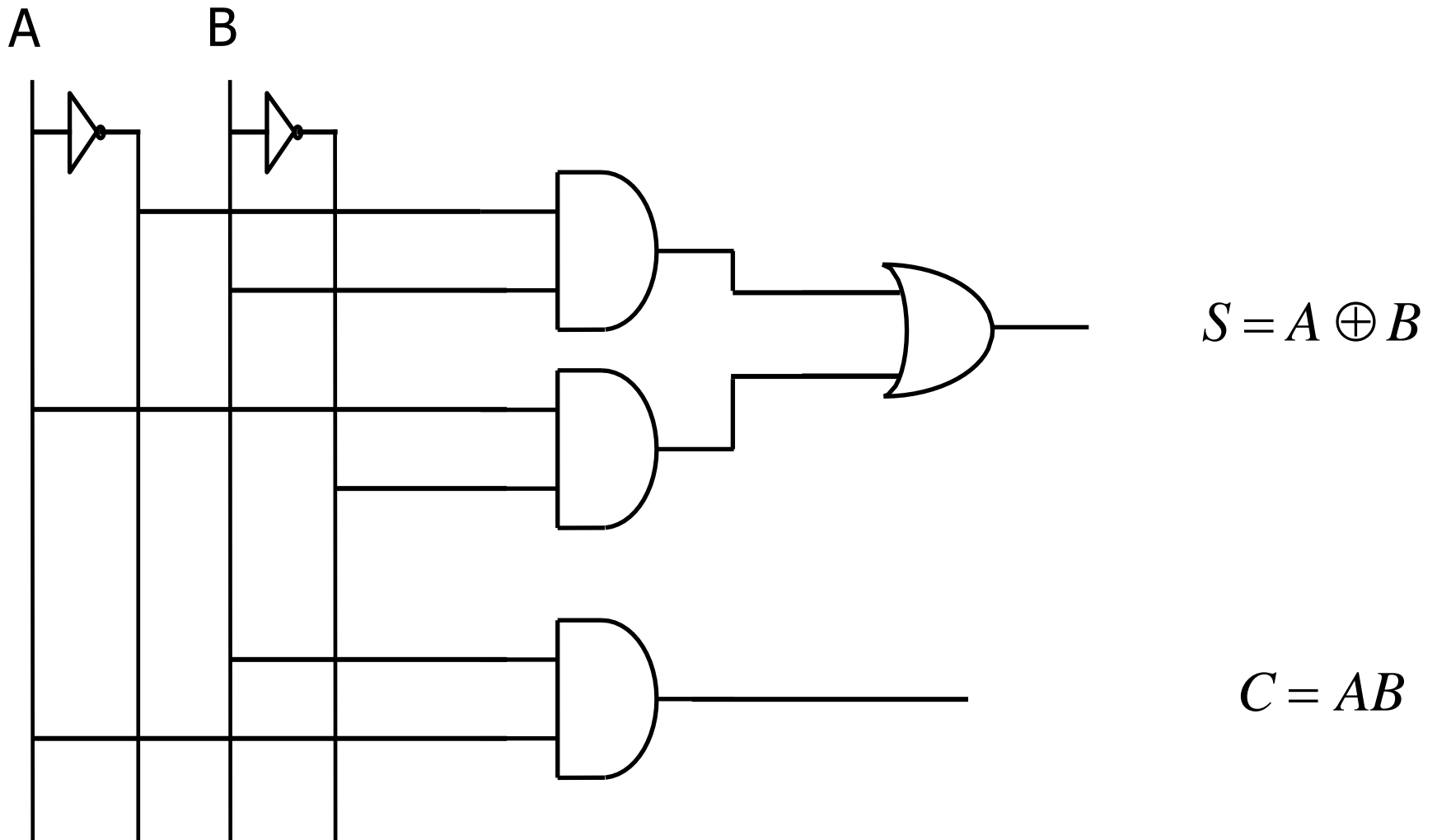
Half Adder

Logic Diagram:



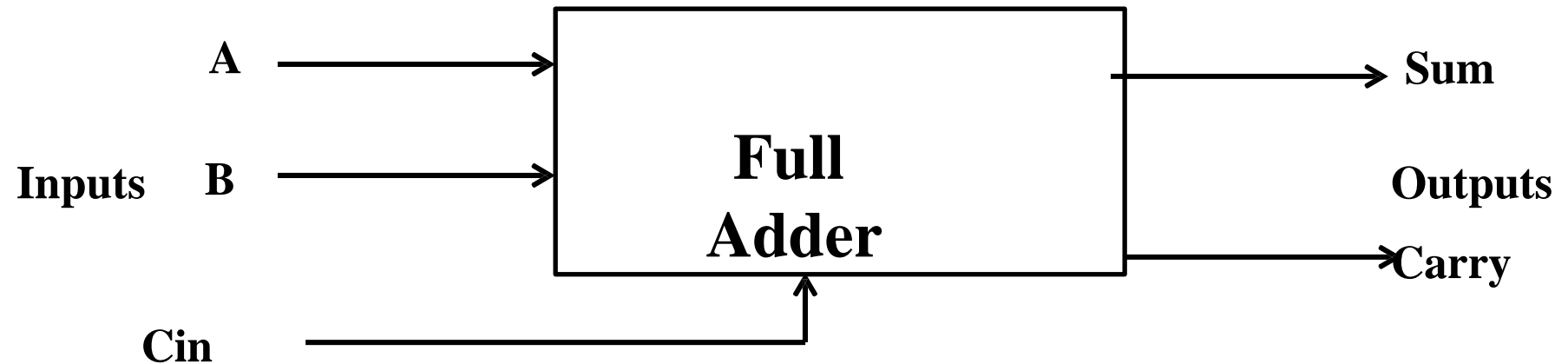
Half Adder

Logic Diagram using Basic Gates:



Full Adder

- ✓ Full adder is a combinational logic circuit with three inputs and two outputs.



Full Adder

Truth Table

Inputs			Outputs	
A	B	Cin	Sum (S)	Carry (C)
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

Full Adder

K-map for Sum Output:

BC A		$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
\overline{A}	0	0	1	0	1
A	1	1	0	1	0

Diagram showing the K-map for Sum Output (S) with groupings and corresponding minterms:

- Group 1 (Top-right and Bottom-left): $\overline{A}\overline{B}C + A\overline{B}\overline{C} = \overline{A}\overline{B}C + A\overline{B}\overline{C}$
- Group 2 (Top-left and Bottom-right): $\overline{A}B\overline{C} + AB\overline{C} = \overline{A}B\overline{C} + AB\overline{C}$
- Group 3 (Top-middle and Bottom-middle): $\overline{A}BC + A\overline{B}C = \overline{A}BC + A\overline{B}C$

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

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

$$S = C(\overline{A}\overline{B} + AB) + \overline{C}(\overline{A}B + A\overline{B})$$

$$\text{Let } \overline{A}B + A\overline{B} = X$$

$$\therefore S = C(\overline{X}) + \overline{C}(X)$$

$$S = C \oplus X$$

$$\text{Let } X = A \oplus B$$

$$\therefore S = C \oplus A \oplus B$$

Full Adder

K-map for Carry Output:

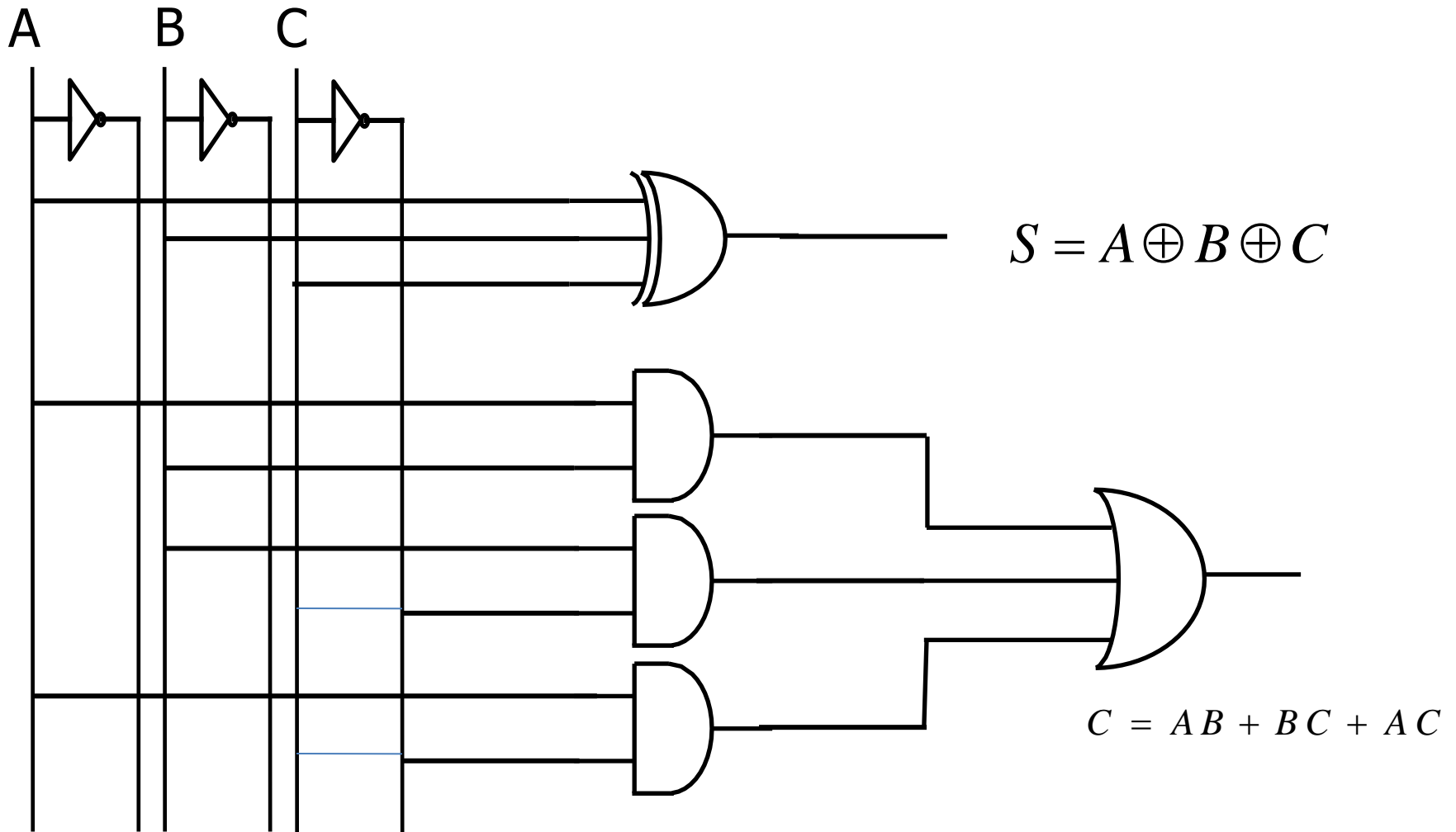
$\begin{array}{c} \text{BC} \\ \text{A} \end{array}$		$\overline{B}\overline{C}$	$\overline{B}C$	BC	$B\overline{C}$
		00	01	11	10
\overline{A}	0	0	0	1	0
A	1	0	1	1	1

Diagram illustrating the K-map for Carry Output (C) of a Full Adder. The K-map is a 2x4 grid with inputs A and BC. The output C is 1 for the following combinations: (A=0, BC=11), (A=1, BC=01), (A=1, BC=11), and (A=1, BC=10). The K-map is grouped into three groups: a vertical group of two cells (A=0, BC=11) and (A=1, BC=11) labeled BC ; a horizontal group of three cells (A=1, BC=01), (A=1, BC=11), and (A=1, BC=10) labeled AB ; and a horizontal group of two cells (A=1, BC=01) and (A=1, BC=11) labeled AC .

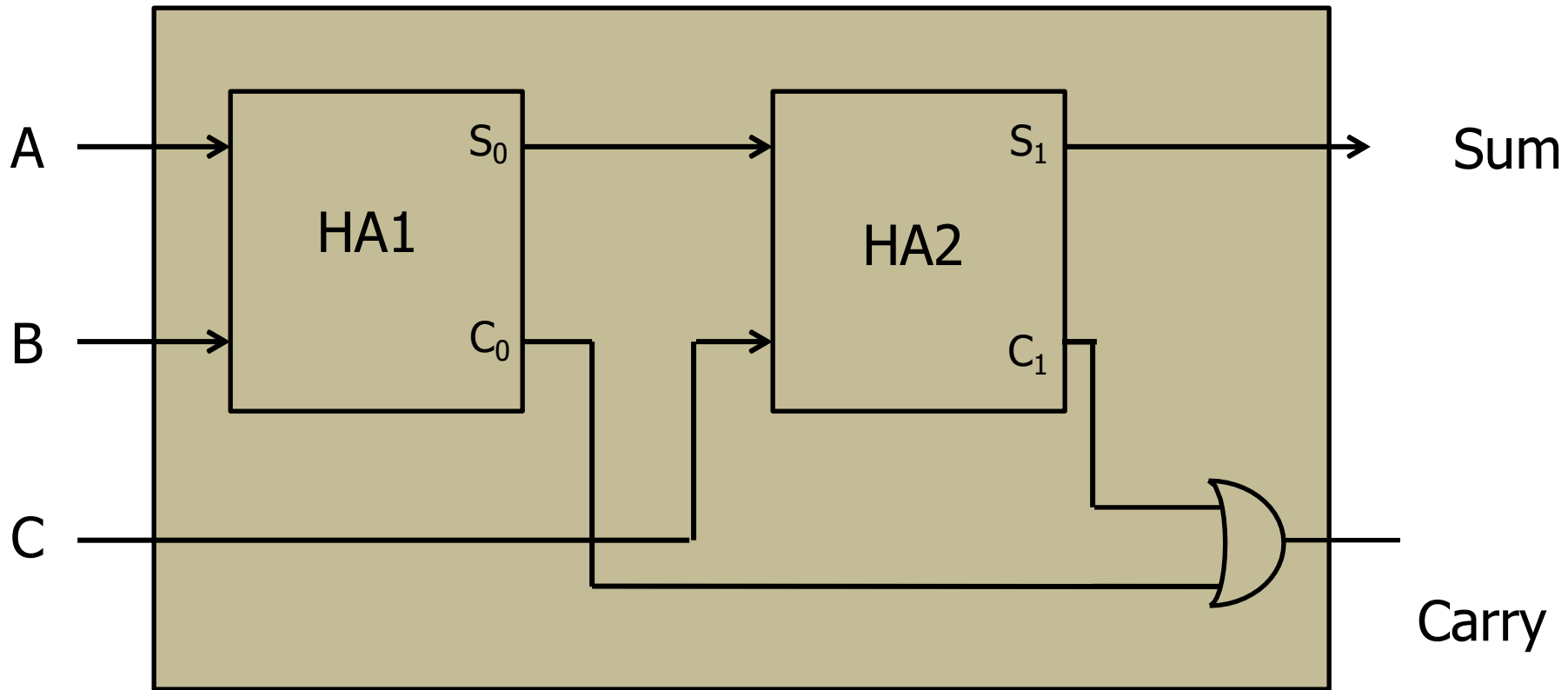
$$C = AB + BC + AC$$

Full Adder

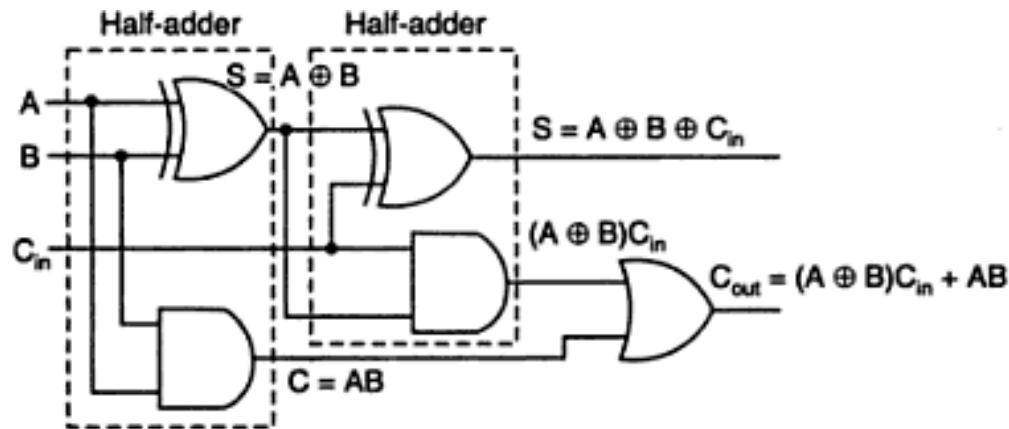
Logic Diagram:



Full Adder using Half Adders

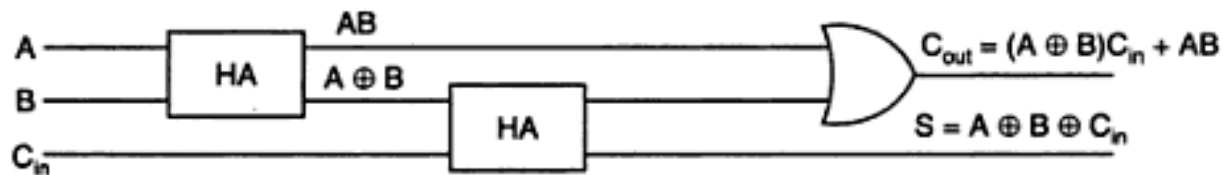


Draw the full adder circuit using two half adders.



Logic diagram of a full-adder using two half-adders.

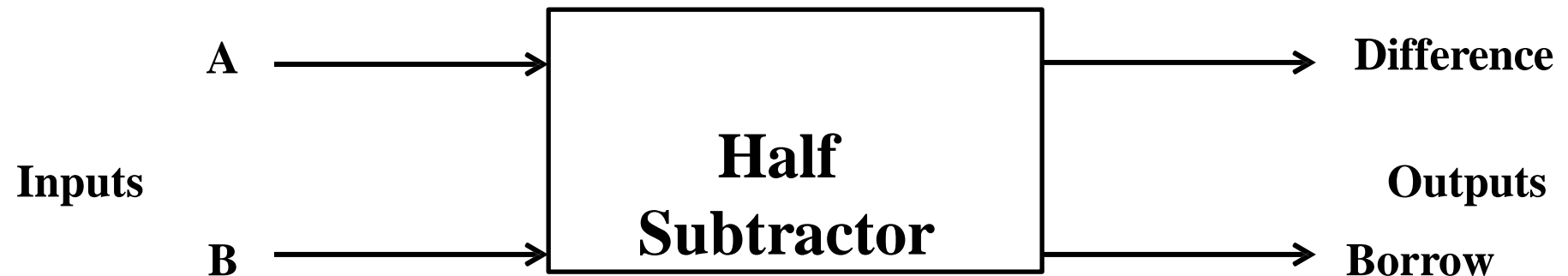
The block diagram of a full-adder using two half-adders is :



Block diagram of a full-adder using two half-adders.

Half Subtractor

- ✓ Half subtractor is a combinational logic circuit with two inputs and two outputs.
- ✓ It is a basic building block for subtraction of two single bit numbers.



Half Subtractor

Truth Table

Input		Output	
A	B	Difference (D)	Borrow (B)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Half Subtractor

K-map for Difference Output:

		B	
		\overline{B}	B
A	\overline{A} 0	0	1
	A 1	1	0

$$D = \overline{A}B + A\overline{B}$$

$$D = A \oplus B$$

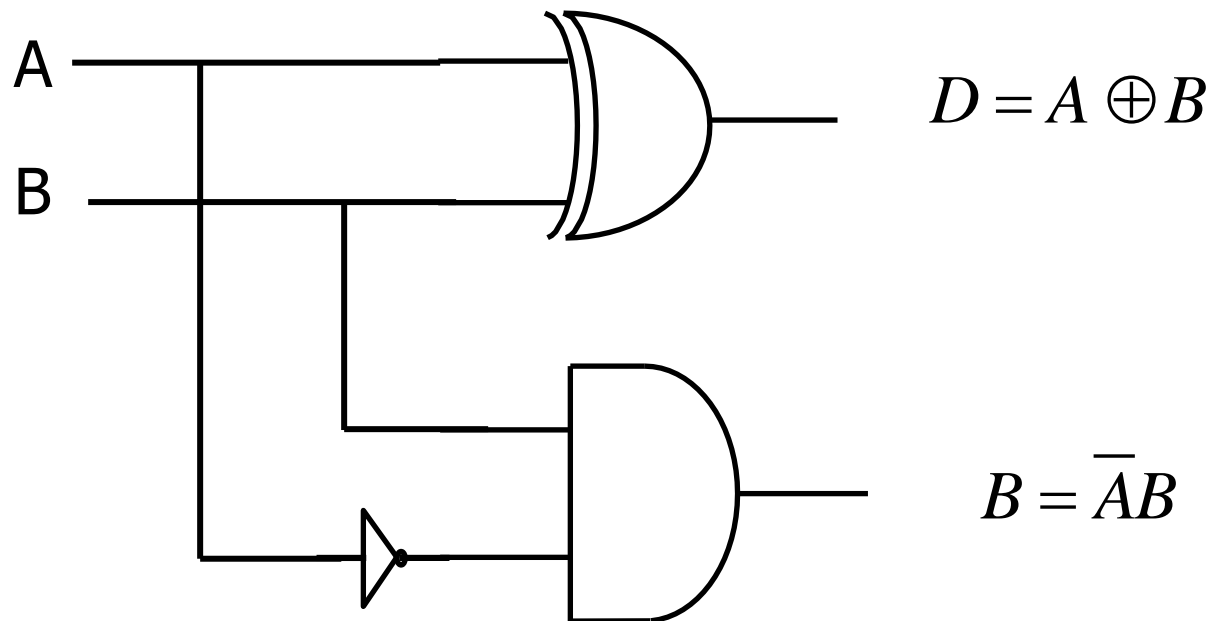
K-map for Borrow Output:

		B	
		\overline{B}	B
A	\overline{A} 0	0	1
	A 1	0	0

$$B = \overline{A}B$$

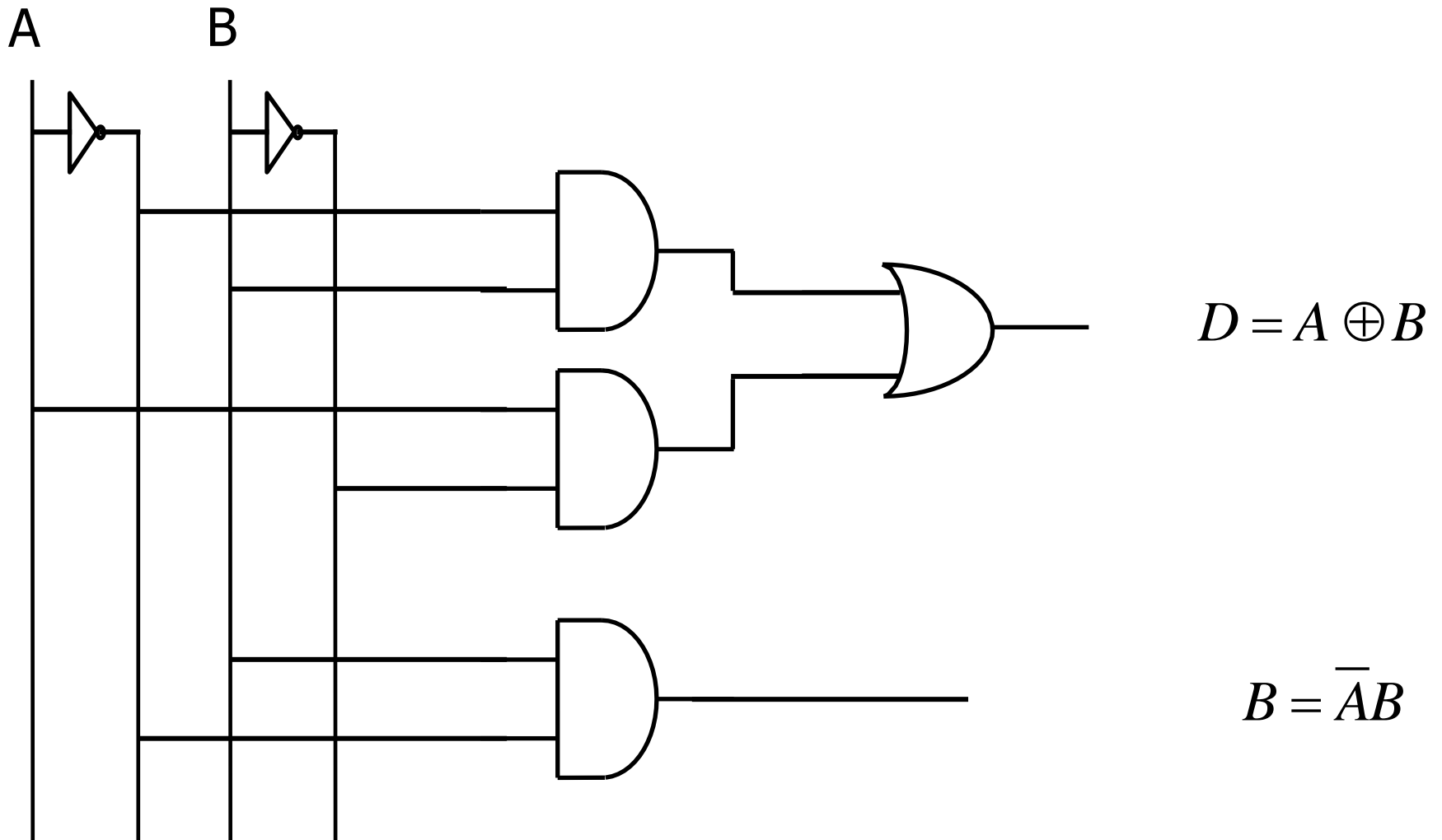
Half Subtractor

Logic Diagram:



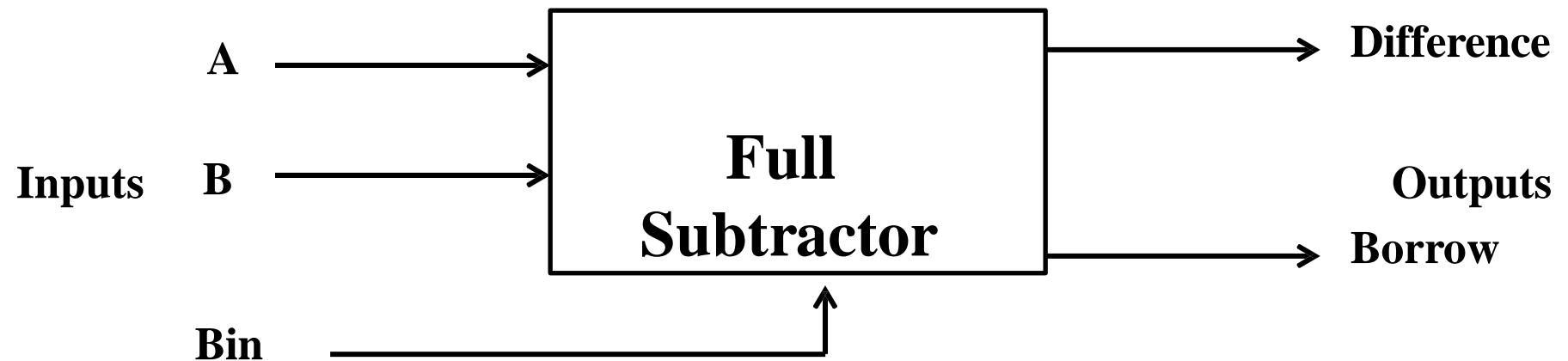
Half Subtractor

Logic Diagram using Basic Gates:



Full Subtractor

- ✓ Full subtractor is a combinational logic circuit with three inputs and two outputs.



Full Subtractor

Truth Table

Inputs			Outputs	
A	B	Bin (C)	Difference (D)	Borrow (B0)
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

Full Subtractor

K-map for Difference Output:

BC A \		$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
\overline{A}	0	0	1	0	1
A	1	1	0	1	0

\downarrow $\overline{A}\overline{B}\overline{C}$ \downarrow $\overline{A}\overline{B}C$ \downarrow ABC \downarrow $\overline{A}B\overline{C}$

$$D = \overline{A}\overline{B}C + \overline{A}B\overline{C} + ABC + A\overline{B}\overline{C}$$

$$D = \overline{A}\overline{B}C + ABC + \overline{A}B\overline{C} + A\overline{B}\overline{C}$$

$$D = C(\overline{A}\overline{B} + AB) + \overline{C}(\overline{A}B + A\overline{B})$$

$$\text{Let } \overline{A}B + A\overline{B} = X$$

$$\therefore D = C(\overline{X}) + \overline{C}(X)$$

$$D = C \oplus X$$

$$\text{Let } X = A \oplus B$$

$$\therefore D = C \oplus A \oplus B$$

Full Subtractor

K-map for Borrow Output:

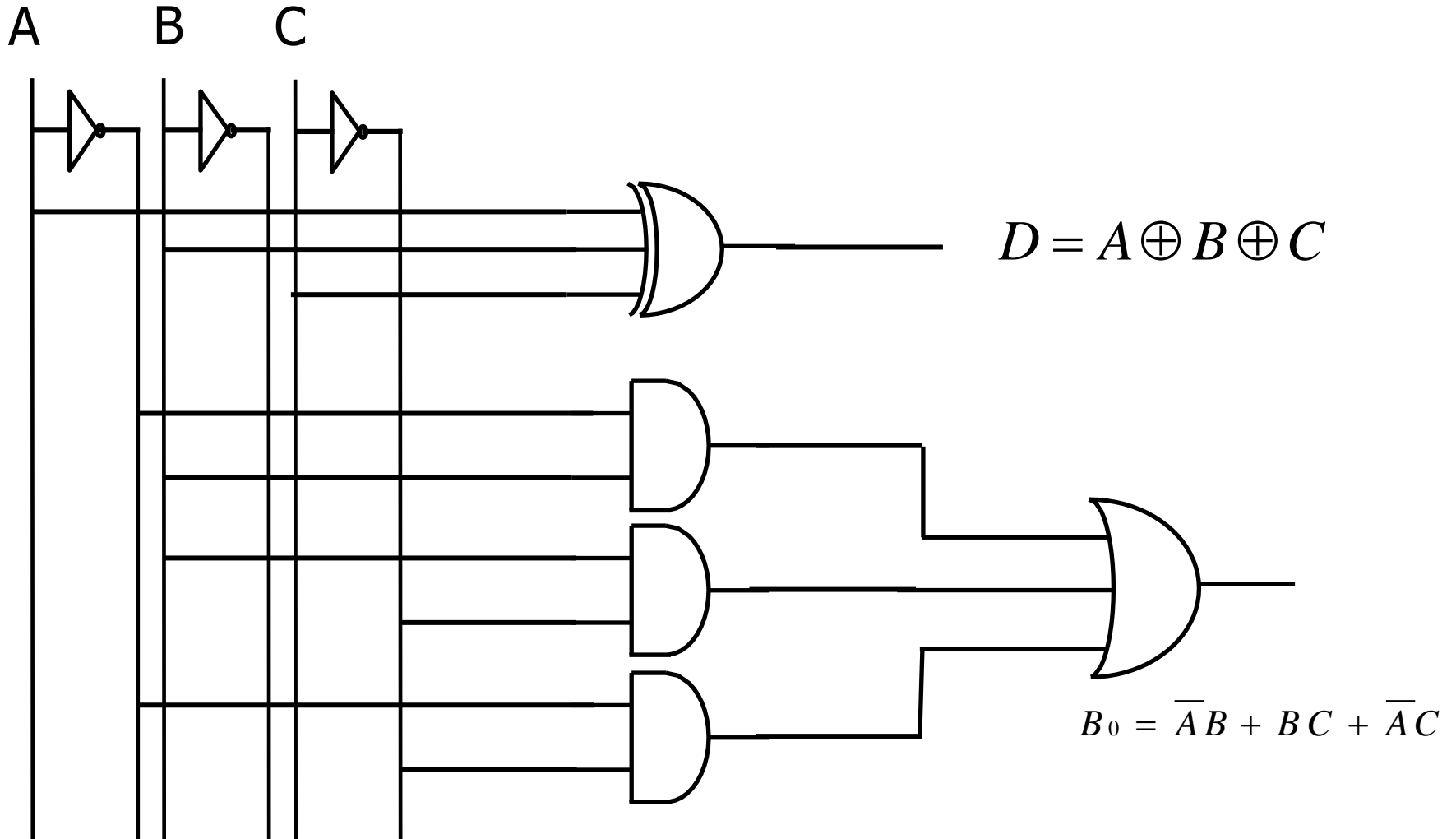
BC A		$\overline{B}\overline{C}$ 00	$\overline{B}C$ 01	BC 11	$B\overline{C}$ 10
\overline{A}	0	0	1	1	1
A	1	0	0	1	0

Diagram illustrating the K-map for Borrow Output (B_0). The K-map is a 2x4 grid with columns labeled $\overline{B}\overline{C}$ (00), $\overline{B}C$ (01), BC (11), and $B\overline{C}$ (10). The rows are labeled \overline{A} (0) and A (1). The values in the cells are: $\overline{A}=0$ row: 0, 1, 1, 1; $A=1$ row: 0, 0, 1, 0. Three groups are circled: a horizontal group of 1s in the $\overline{A}=0$ row (labeled $\overline{A}C$), a vertical group of 1s in the BC column (labeled BC), and a group of 1s in the $\overline{A}=0$ row and $A=1$ row for the $\overline{B}C$ and $B\overline{C}$ columns (labeled $\overline{A}B$).

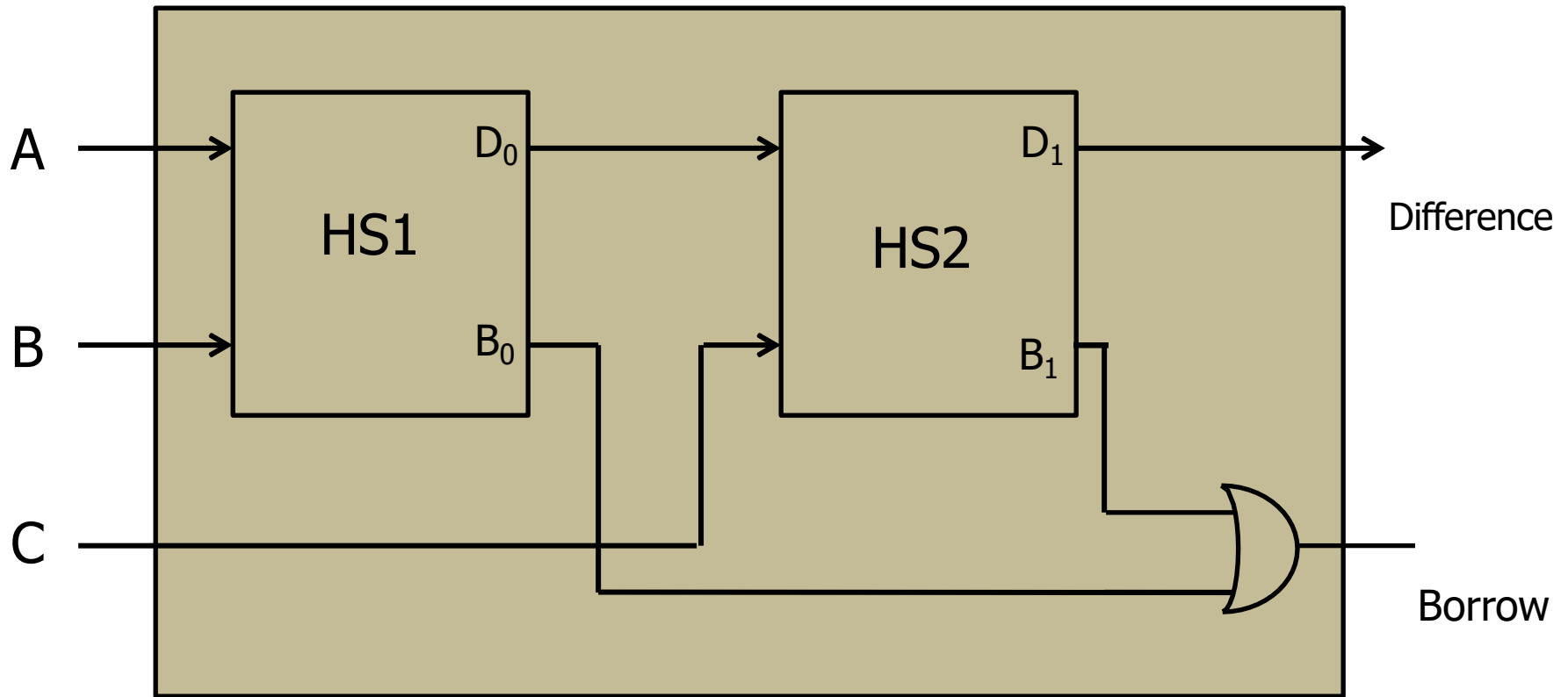
$$B_0 = \overline{A}B + BC + \overline{A}C$$

Full Subtractor

Logic Diagram:

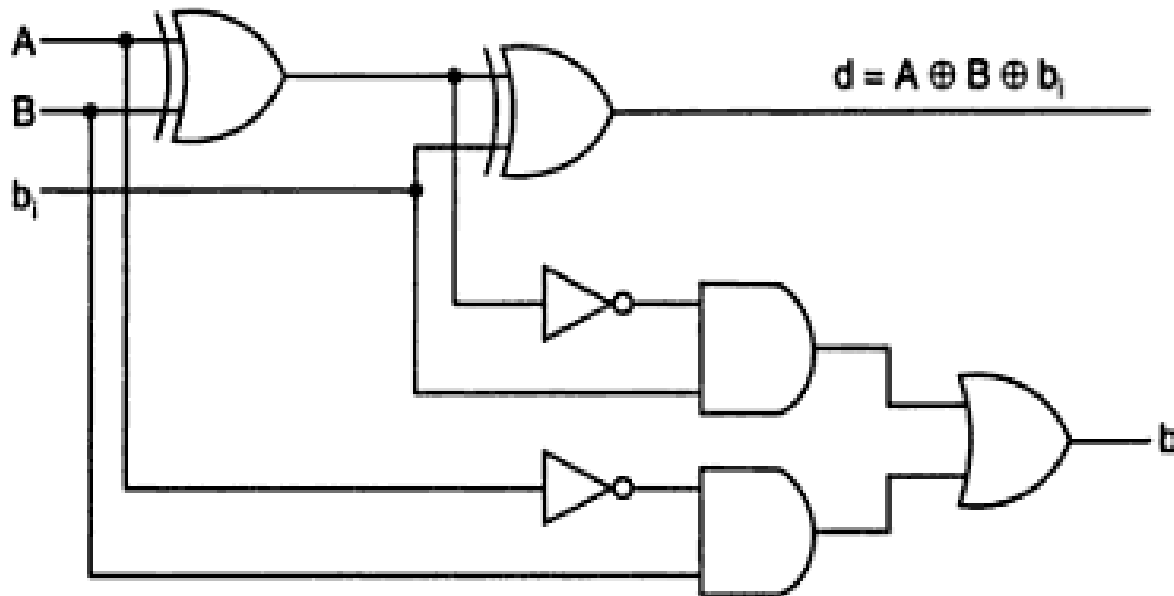


Full Subtractor using Half Subtractor



Draw a full subtractor circuit using two half subtractors.

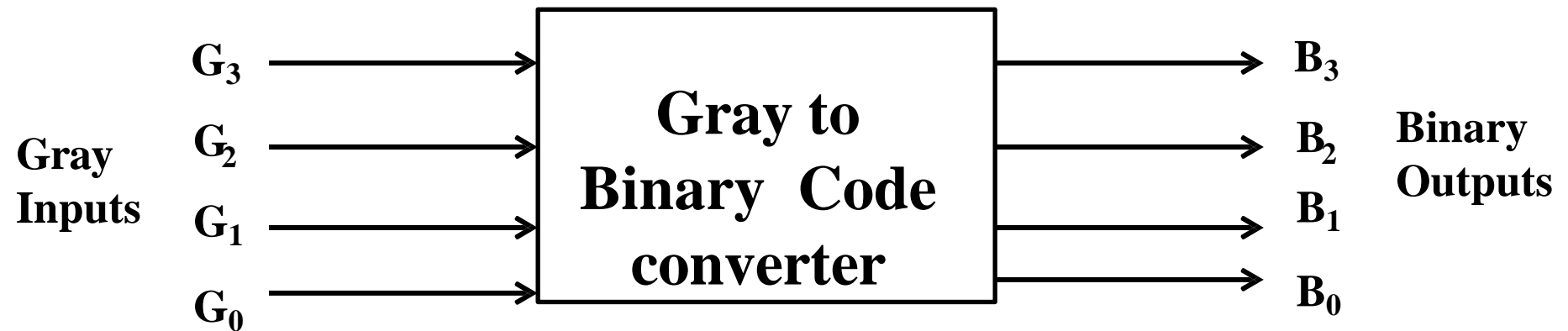
The circuit diagram is as:



Logic diagram of a full-subtractor.

Design of Gray to Binary Code Converter

Block Diagram:



Design of Gray to Binary Code Converter

Truth Table :

Gray Inputs			Binary Outputs			
G ₂	G ₁	G ₀	B ₃	B ₂	B ₁	B ₀
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	0	1	1
0	1	1	0	0	1	0
1	0	0	0	1	1	0
1	0	1	0	1	1	1
1	1	0	0	1	0	1
1	1	1	0	1	0	0

Gray Inputs				Binary Outputs			
G ₃	G ₂	G ₁	G ₀	B ₃	B ₂	B ₁	B ₀
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Design of Gray to Binary Code Converter

K-map for B_0 :

		$G_1 G_0$			
		$\overline{G_1} \overline{G_0}$	$\overline{G_1} G_0$	$G_1 G_0$	$G_1 \overline{G_0}$
$G_3 G_2$	$\overline{G_3} \overline{G_2}$	00	01	11	10
	$\overline{G_3} G_2$	0 ⁰	1 ¹	0 ³	1 ²
	$G_3 \overline{G_2}$	1 ⁴	0 ⁵	1 ⁷	0 ⁶
	$G_3 G_2$	0 ¹²	1 ¹³	0 ¹⁵	1 ¹⁴
		$\overline{G_3} \overline{G_2}$	$\overline{G_3} G_2$	$G_3 \overline{G_2}$	$G_3 G_2$
		00	01	11	10
		0	1	0	1
		1	0	1	0
		0	1	0	1
		1	0	1	0

$$B_0 = \overline{G_3} \overline{G_2} \overline{G_1} G_0 + \overline{G_3} \overline{G_2} G_1 \overline{G_0} + \overline{G_3} G_2 \overline{G_1} \overline{G_0} + \overline{G_3} G_2 G_1 G_0 \\ + G_3 \overline{G_2} \overline{G_1} G_0 + G_3 \overline{G_2} G_1 \overline{G_0} + G_3 G_2 \overline{G_1} G_0 + G_3 G_2 G_1 \overline{G_0}$$

$$B_0 = G_3 \oplus G_2 \oplus G_1 \oplus G_0$$

Design of Gray to Binary Code Converter

K-map for B_1 :

		G_1G_0				
		$\overline{G_1}\overline{G_0}$	$\overline{G_1}G_0$	G_1G_0	$G_1\overline{G_0}$	
G_3G_2	$\overline{G_3}\overline{G_2}$	00	01	11	10	
	00	0 ⁰	0 ¹	1 ³	1 ²	
	$\overline{G_3}G_2$	01	1 ⁴	1 ⁵	0 ⁷	0 ⁶
	11	0 ¹²	0 ¹³	1 ¹⁵	1 ¹⁴	
	$G_3\overline{G_2}$	10	1 ⁸	1 ⁹	0 ¹¹	0 ¹⁰

$$B_1 = \overline{G_3}\overline{G_2}G_1 + \overline{G_3}G_2\overline{G_1} + G_3G_2G_1 + G_3\overline{G_2}\overline{G_1}$$

$$B_1 = G_3 \oplus G_2 \oplus G_1$$

Design of Gray to Binary Code Converter

K-map for B_2 :

		G_1G_0			
		$\overline{G_1}\overline{G_0}$	$\overline{G_1}G_0$	G_1G_0	$G_1\overline{G_0}$
G_3G_2	$\overline{G_3}\overline{G_2}$	00	01	11	10
	$\overline{G_3}G_2$	00	01	11	10
G_3G_2	$\overline{G_3}\overline{G_2}$	0	1	3	2
	$\overline{G_3}G_2$	0	0	0	0
G_3G_2	$\overline{G_3}\overline{G_2}$	4	5	7	6
	$\overline{G_3}G_2$	1	1	1	1
G_3G_2	$\overline{G_3}\overline{G_2}$	12	13	15	14
	$\overline{G_3}G_2$	0	0	0	0
G_3G_2	$\overline{G_3}\overline{G_2}$	8	9	11	10
	$\overline{G_3}G_2$	1	1	1	1

$$B_2 = \overline{G_3}G_2 + G_3\overline{G_2}$$

$$B_1 = G_3 \oplus G_2$$

Design of Gray to Binary Code Converter

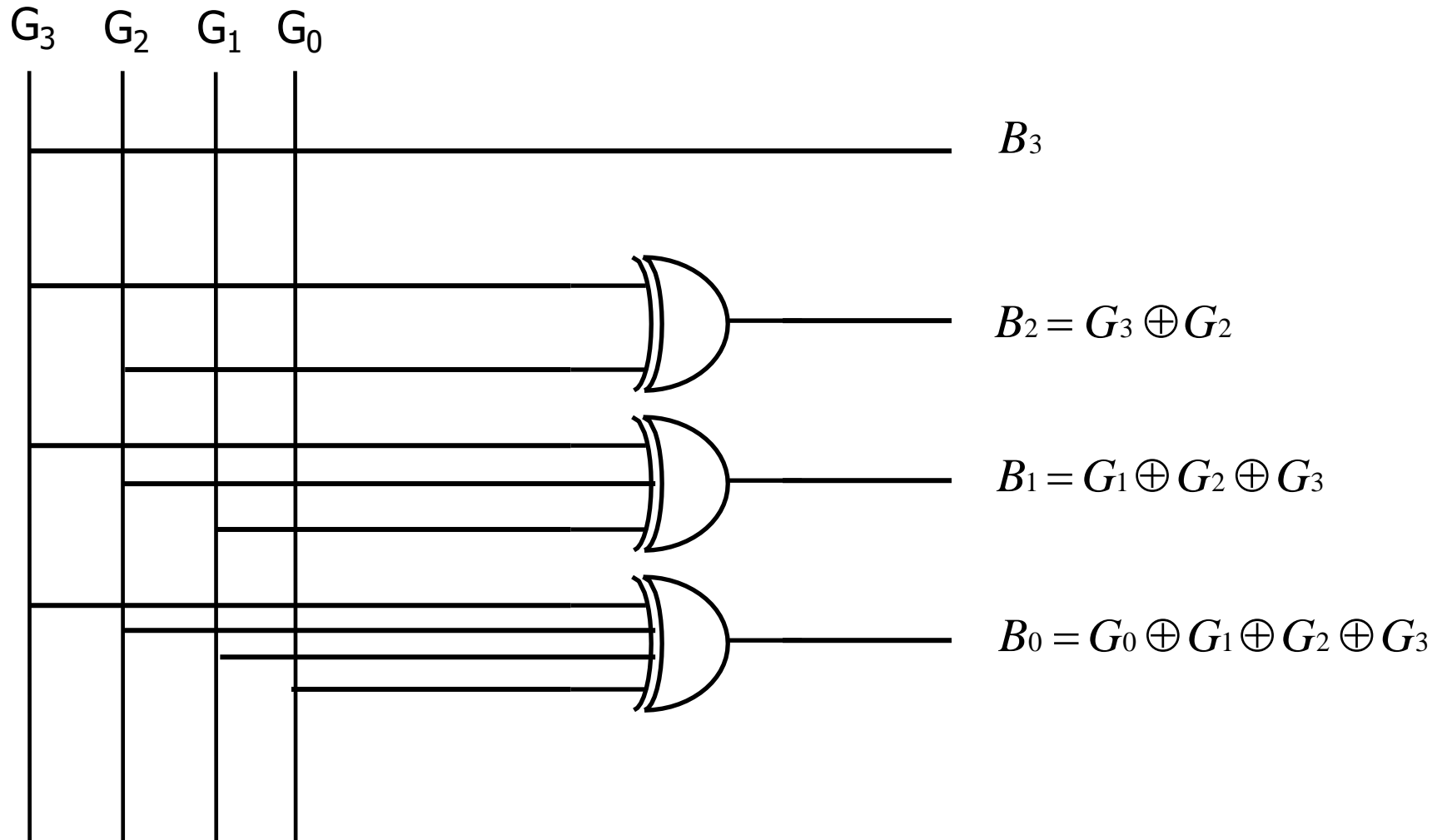
K-map for B_3 :

G_3G_2 \ G_1G_0		$\overline{G_1}\overline{G_0}$	$\overline{G_1}G_0$	G_1G_0	$G_1\overline{G_0}$
		00	01	11	10
$\overline{G_3}\overline{G_2}$	00	0 ⁰	0 ¹	0 ³	0 ²
$\overline{G_3}G_2$	01	0 ⁴	0 ⁵	0 ⁷	0 ⁶
$G_3\overline{G_2}$	11	1 ¹²	1 ¹³	1 ¹⁵	1 ¹⁴
G_3G_2	10	1 ⁸	1 ⁹	1 ¹¹	1 ¹⁰

$$B_3 = G_3$$

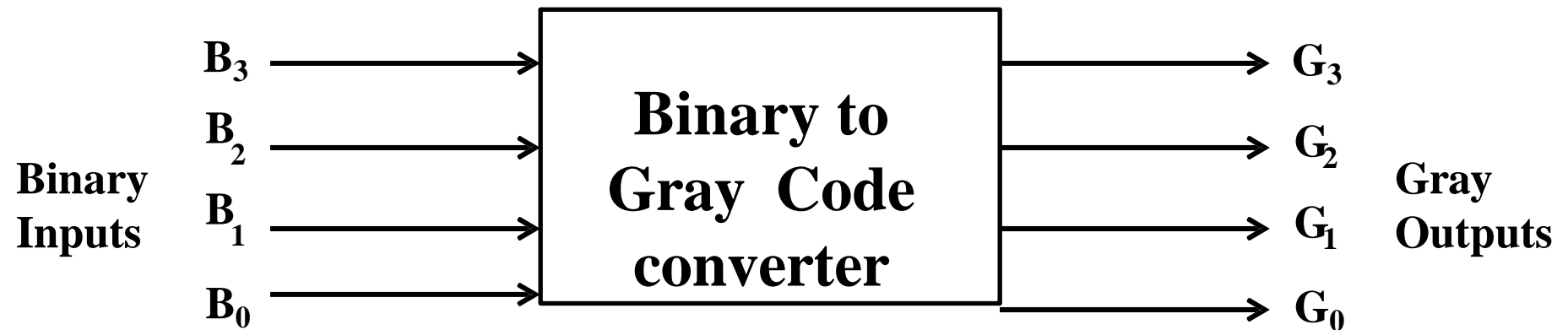
Design of Gray to Binary Code Converter

Logic Diagram:



Design of Binary to Gray Code Converter

Block Diagram:



Design of Binary to Gray Code Converter

Truth Table :

Binary Inputs				Gray Outputs			
B ₃	B ₂	B ₁	B ₀	G ₃	G ₂	G ₁	G ₀
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0

Binary Inputs				Gray Outputs			
B ₃	B ₂	B ₁	B ₀	G ₃	G ₂	G ₁	G ₀
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Design of Binary to Gray Code Converter

K-map for G_0 :

B_3B_2 \ B_1B_0		$\overline{B_1}\overline{B_0}$	$\overline{B_1}B_0$	$B_1\overline{B_0}$	B_1B_0
		00	01	11	10
$\overline{B_3}\overline{B_2}$ 00	0	0	1	0	1
$\overline{B_3}B_2$ 01	4	0	1	0	1
$B_3\overline{B_2}$ 11	12	0	1	0	1
B_3B_2 10	8	0	1	0	1

Two groups of 1s are circled in the K-map, each containing four 1s. The first group is labeled $\overline{B_1}B_0$ and the second group is labeled $B_1\overline{B_0}$.

$$G_0 = \overline{B_1}B_0 + B_1\overline{B_0}$$

$$\therefore G_0 = B_0 \oplus B_1$$

Design of Binary to Gray Code Converter

K-map for G_1 :

B_3B_2 \ B_1B_0		$\overline{B_1}\overline{B_0}$	$\overline{B_1}B_0$	$B_1\overline{B_0}$	B_1B_0
		00	01	11	10
$\overline{B_3}\overline{B_2}$ 00	0	0	1	1	2
$\overline{B_3}B_2$ 01	4	1	1	0	6
$B_3\overline{B_2}$ 11	12	1	1	0	14
B_3B_2 10	8	0	0	1	10

Group 1 (Top-Left): $B_2\overline{B_1}$

Group 2 (Top-Right): $\overline{B_2}B_1$

$$G_1 = \overline{B_2}B_1 + B_2\overline{B_1}$$

$$\therefore G_1 = B_2 \oplus B_1$$

Design of Binary to Gray Code Converter

K-map for G_2 :

B_1B_0 B_3B_2		$\overline{B_1}\overline{B_0}$	$\overline{B_1}B_0$	B_1B_0	$B_1\overline{B_0}$
		00	01	11	10
$\overline{B_3}\overline{B_2}$	00	0 ⁰	0 ¹	0 ³	0 ²
$\overline{B_3}B_2$	01	1 ⁴	1 ⁵	1 ⁷	1 ⁶
B_3B_2	11	0 ¹²	0 ¹³	0 ¹⁵	0 ¹⁴
$B_3\overline{B_2}$	10	1 ⁸	1 ⁹	1 ¹¹	1 ¹⁰

$B_3\overline{B_2}$

$\overline{B_3}B_2$

$$G_2 = \overline{B_3}B_2 + B_3\overline{B_2}$$

$$\therefore G_2 = B_3 \oplus B_2$$

Design of Binary to Gray Code Converter

K-map for G_3 :

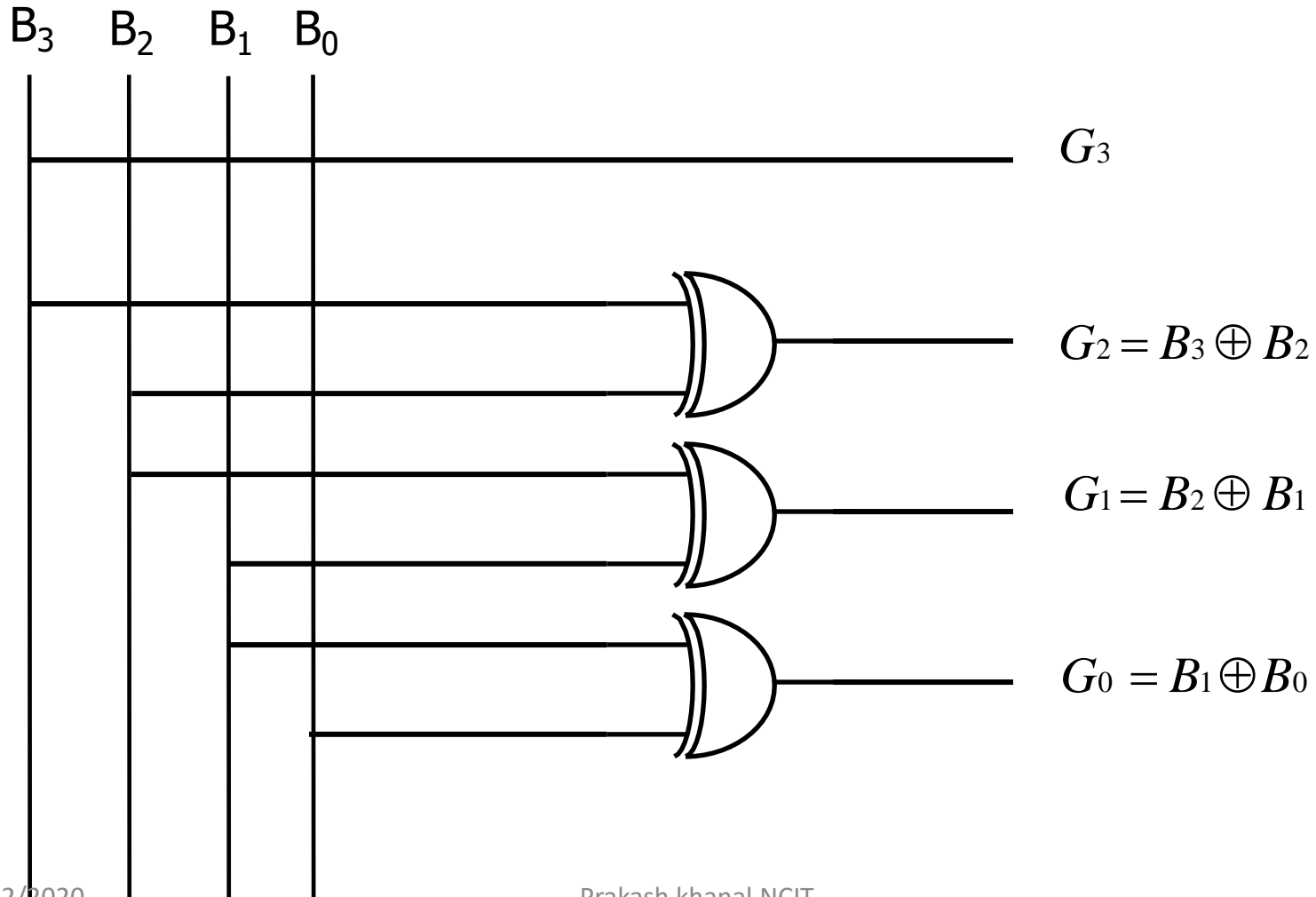
B_3B_2 \ B_1B_0		$\overline{B_1}\overline{B_0}$	$\overline{B_1}B_0$	$B_1\overline{B_0}$	B_1B_0
		00	01	11	10
$\overline{B_3}\overline{B_2}$ 00	0	0	1	3	2
$\overline{B_3}B_2$ 01	4	0	5	7	6
$B_3\overline{B_2}$ 11	12	1	13	15	14
B_3B_2 10	8	1	9	11	10

B_3

$$G_3 = B_3$$

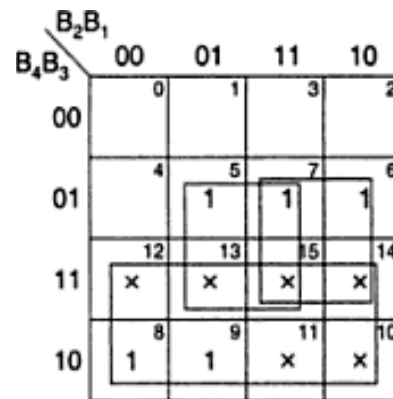
Design of Binary to Gray Code Converter

Logic Diagram:



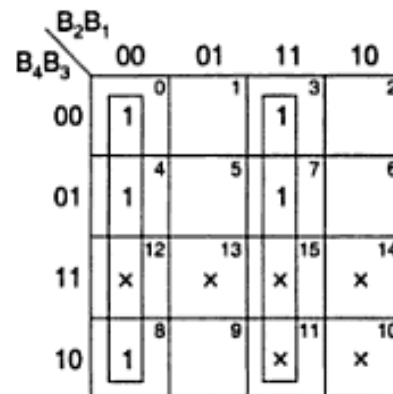
Design of a 4-bit BCD to XS-3 code converter:

8421 code				XS-3 code			
B_4	B_3	B_2	B_1	X_4	X_3	X_2	X_1
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0



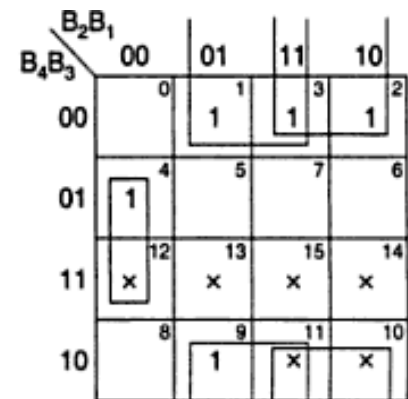
$$X_4 = B_4 + B_3B_2 + B_3B_1$$

K-map for X_4



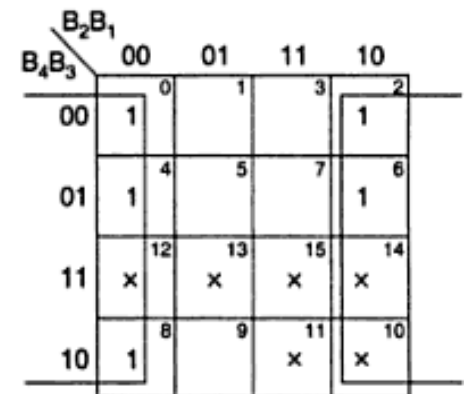
$$X_2 = \bar{B}_2\bar{B}_1 + B_2B_1$$

K-map for X_2



$$X_3 = B_3\bar{B}_2\bar{B}_1 + \bar{B}_3B_1 + \bar{B}_3B_2$$

K-map for X_3



$$X_1 = \bar{B}_1$$

K-map for X_1

(c) K-maps

4-bit BCD-to-XS-3 code converter.

The o/p functions and logic diagram is as:

The minimal expressions are

$$X_4 = B_4 + B_3B_2 + B_3B_1$$

$$X_3 = B_3\bar{B}_2\bar{B}_1 + \bar{B}_3B_1 + \bar{B}_3B_2$$

$$X_2 = \bar{B}_2\bar{B}_1 + B_2B_1$$

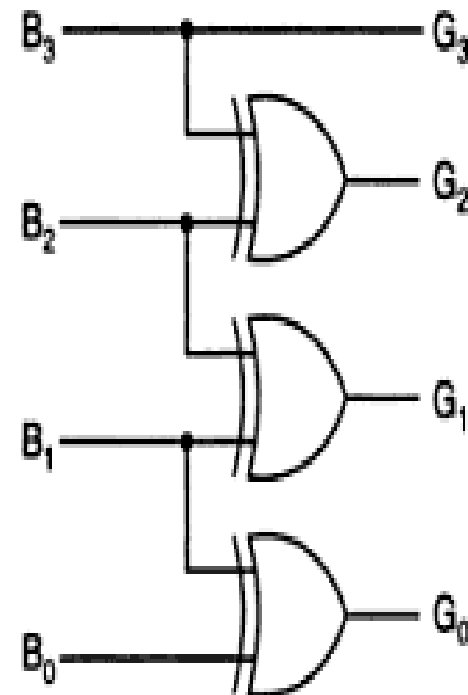
$$X_1 = \bar{B}_1$$

- Draw the logic diagram yourself.

Design of a BCD to gray code converter:

BCD code				Gray code			
B ₃	B ₂	B ₁	B ₀	G ₃	G ₂	G ₁	G ₀
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1

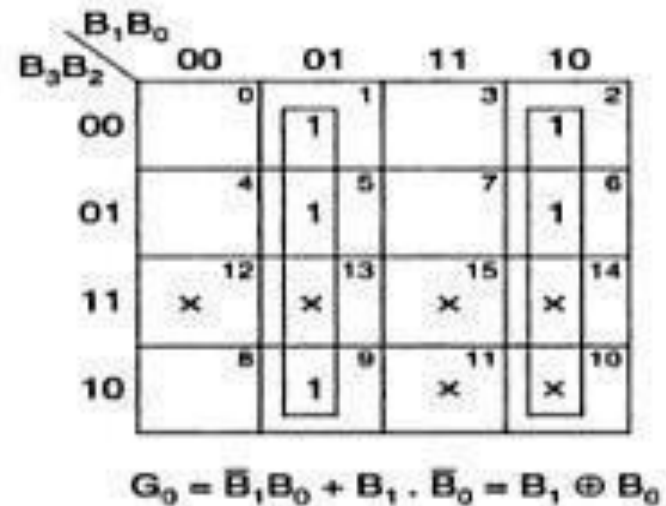
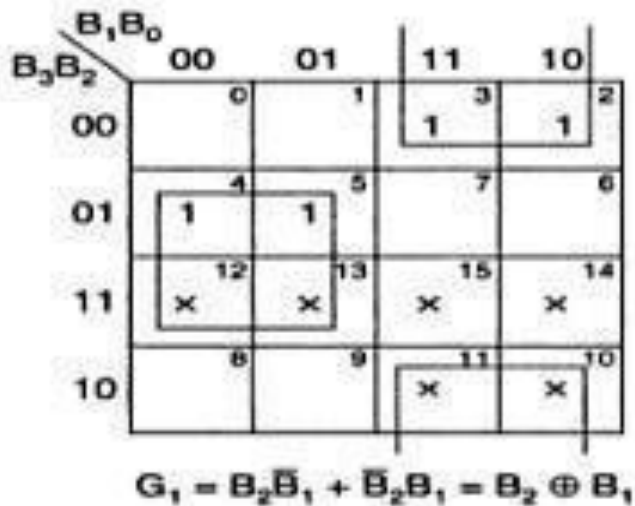
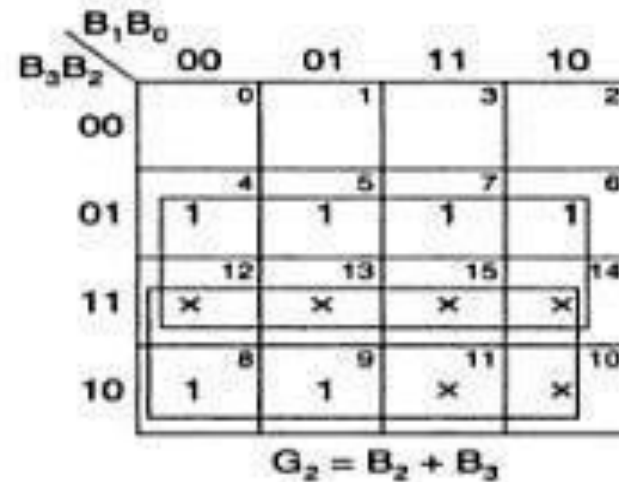
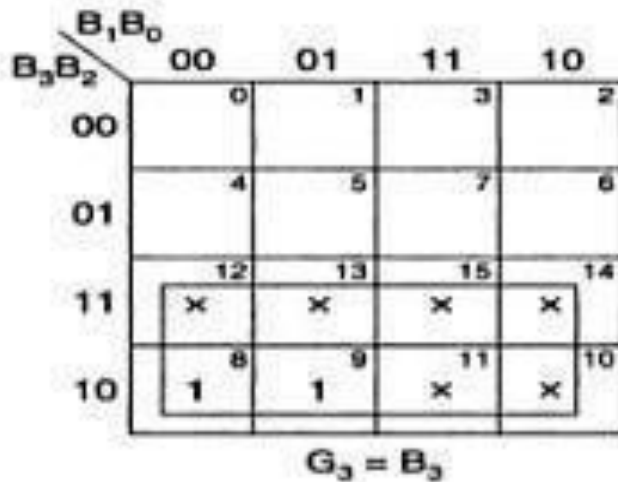
(a) BCD-to-Gray code conversion table



(b) Logic diagram

BCD-to-Gray code converter.

Using K-map,



K-maps for a BCD-to-Gray code converter.

Design of a Combinational circuit to produce the 2's complement of a 4-bit binary number:

Input				Output			
A	B	C	D	E	F	G	H
0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1
0	0	1	0	1	1	1	0
0	0	1	1	1	1	0	1
0	1	0	0	1	1	0	0
0	1	0	1	1	0	1	1
0	1	1	0	1	0	1	0
0	1	1	1	1	0	0	1
1	0	0	0	1	0	0	0
1	0	0	1	0	1	1	1
1	0	1	0	0	1	1	0
1	0	1	1	0	1	0	1
1	1	0	0	0	1	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	0	1	0
1	1	1	1	0	0	0	1

(a) Conversion table

Conversion table and K-maps for the circuit

- Simplify the o/p and draw the logic diagram yourself.

K Map for:

E: AB

	00	01	11	10
00		1	1	1
01	1	1	1	1
11				
10	1			

$\rightarrow \bar{A}B$
 $\rightarrow \bar{A}C$
 $\rightarrow \bar{A}D$
 \downarrow
 $A\bar{B}\bar{C}\bar{D}$

$$\therefore E = \bar{A}B + \bar{A}C + \bar{A}D + A\bar{B}\bar{C}\bar{D}$$

F:

	00	01	11	10
00		1	1	1
01	1			
11	1			
10		1	1	1

\downarrow
 $B\bar{C}D$ $\bar{B}D$ $\bar{B}\bar{C}$

$$\therefore F = \bar{B}D + \bar{B}\bar{C} + B\bar{C}D$$

G:

	00	01	11	10
00		1		1
01		1		1
11		1		1
10		1		1

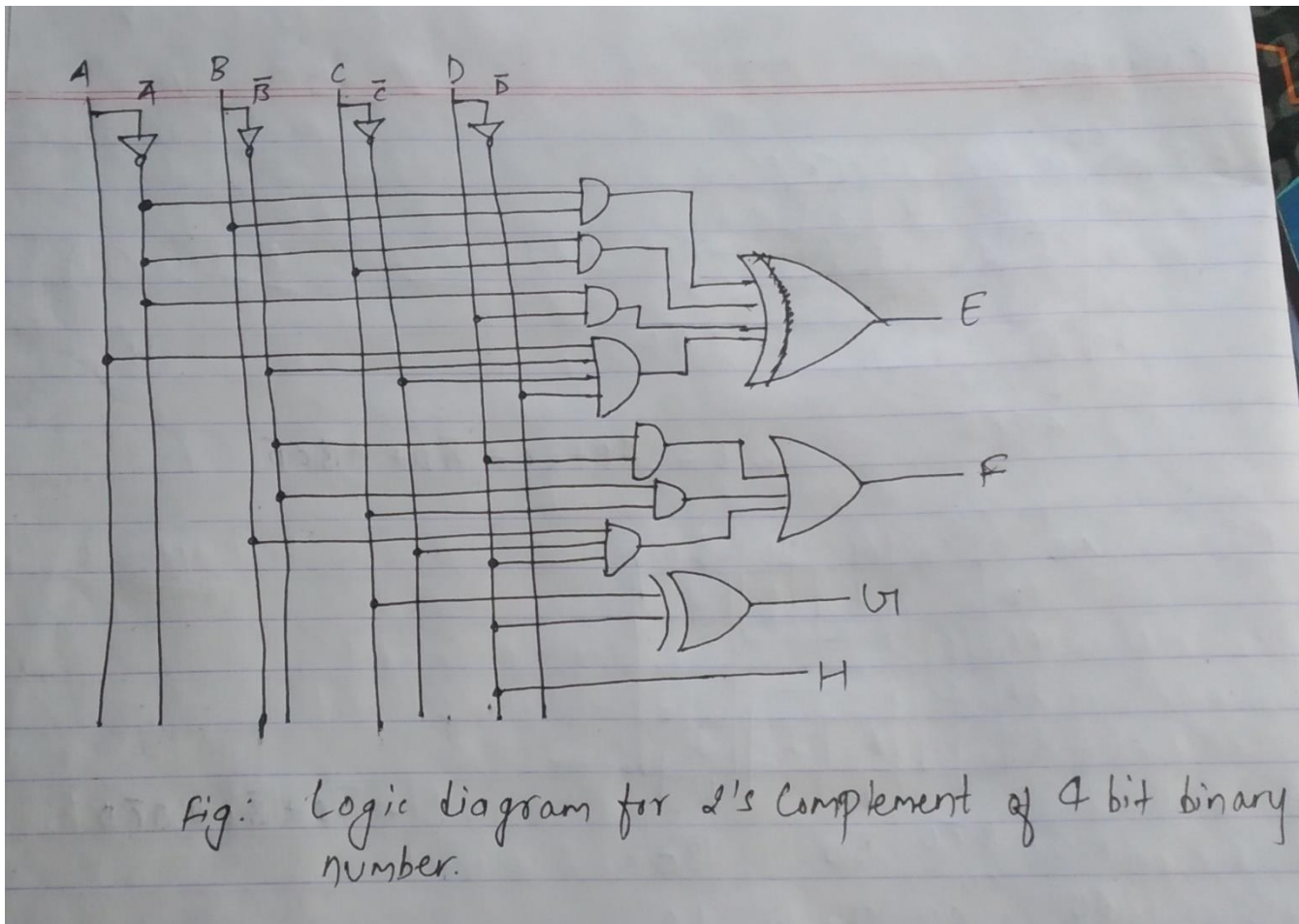
\downarrow
 $\bar{C}D$ CD

$$\therefore G = \bar{C}D + CD = C \oplus D$$

H:

	00	01	11	10
00		1	1	
01		1	1	
11		1	1	
10		1	1	

$$\therefore H = D$$



Parity method for error detection and correction:

- Parity bit is an extra bit included with a binary message to make the number of 1's either odd or even.
- The message including the parity bit, is transmitted and checked at the receiving end for errors.
- An error is detected if the checked parity does not correspond to the one transmitted.
- The circuit that generates the parity bit in the transmitter is called a parity generator, the circuit that checks the parity in the receiver is called a parity checker.

Design a circuit for parity generator for 3 bit message with even and odd parity:

X	y	z	Odd Parity bit generated(p)
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Simplify the parity using k map and design the circuit.

Design a circuit for odd parity check:

Four-bits received				Parity-error check
x	y	z	P	C
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Simplify the k map for output and draw the logic daigram.

Four-bits received				Parity-error check
x	y	z	P	C
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1