Formulae

FRL Self Duality

- How many self-dual functions can be made with n variables?
    - Total number of combinations: 2^n
    - Total number of Boolean functions: 2^{2^n}
    - Total number of self-dual functions: $2^{2^{n-1}}$

Gates

Properties of Gates:

- Idempotent: $A.A = A$
- Commutative: $A.B = B.A$
- Associative: $(A + B) + C = A + (B + C)$

[1/0 = Doesn't have/has property]

Gates	Symbol	Idempotent/Closure	Commutative	Associative
NOT	¬	0	NA	NA
AND	.	1	1	1
OR	+	1	1	1
NAND	↑	0	1	0
NOR	↓	0	1	0
XOR	⊕	0	1	1
XNOR	⊙	0	1	1

Types of Gates

- Basic Gates: The most basic of all: AND, OR, NOT
- Universal Gates: Combination of basic gates, can be used to make any other gate: NAND (AND + NOT), NOR (OR + NOT)
- Arithmetic Gates: Used to perform arithmetic operations: XOR, XNOR
- Evaluation:
    - AND: **1** if both A & B are 1, otherwise **0**.
    - OR: **1** if either of A or B are 1, otherwise **0**.
    - NOR: Negation of OR, **1** if A OR B is 0, otherwise **0**.
    - NAND: Negation or AND, **1** if A AND B is 0, otherwise **0**.
    - XOR: **1** if both A & B are different, otherwise **0**.
    - XNOR: **1** if both A & B are same, otherwise **0**.

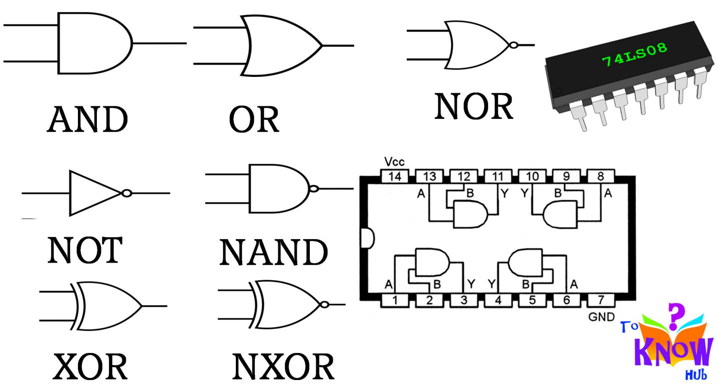


Image taken from here

A	B	AND	OR	NOR	NAND	XOR	XNOR
A	B	A.B	A+B	¬(A.B)	¬(A+B)	$A \oplus B$	$A \odot B$
0	0	0	0	1	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	1	0	0	0	1

Universal Gates: NAND and NOR

- Logic: To implement OR gate, we need 3 NAND gates or 2 NOR gates.

Gates	NAND	NOR
NOT	1	1
AND	2	3
OR	3	2
XOR	4	5
XNOR	5	4

Arithmetic Gates (XOR, XNOR)

XOR Gate

- If 2 inputs are A & B, A ⊕ B = $A\bar{B} + \bar{A}B$
- If both inputs are same, output will be `0`, otherwise `1`.
- Truth Table:

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

- Examples:
  i. $A \oplus A = 0$
  ii. $A \oplus \bar{A} = 1$
  iii. $A \oplus 0 = A$ (Exchange property: In `(i)`, put `0` in LHS & `A` in RHS)
  iv. $A \oplus 1 = \bar{A}$ (Exchange property: In `(ii)`, put `1` in LHS & `$\bar A$` in RHS)
  v. $A \oplus A \oplus A \oplus A \oplus A \oplus A ... n\ times$:
    - If `n` is even (for example $n = 4$), $A \oplus A \oplus A \oplus A = 0 \oplus 0 = 0$

- If `n` is odd (for example $n=3$), $A \oplus A \oplus A = 0 \oplus A = A$

- XOR Gate can be used as buffer/inverter: Since $A \oplus 0 = A$ & $A \oplus 1 = \bar{A}$, we can give an input `A` & use the 2nd input as control. If the control is 0, output is same as input, otherwise output is inverted.

XNOR Gate

- Negation (¬) of XOR Gate.
- If 2 inputs are A & B, A ⊙ B = $AB + \bar{A}\bar{B}$
- If both inputs are same, output will be `1`, otherwise `0`.
- Truth Table:

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

- Examples:
  i. $A \odot A = 1$
  ii. $A \odot \bar{A} = 0$
  iii. $A \odot 0 = \bar{A}$ (Exchange property: In `(i)`, put `0` in LHS & `A` in RHS)
  iv. $A \odot 1 = A$ (Exchange property: In `(ii)`, put `1` in LHS & `$\bar A$` in RHS)
  v. $A \odot A \odot A \odot A \odot A \odot A ... n\ times$:
    - If `n` is even (for example $n=4$), $A \odot A \odot A \odot A = 1 \odot 1 = 1$
    - If `n` is odd (for example $n=3$), $A \odot A \odot A = 1 \oplus A = A$
- XNOR Gate can be used as buffer/inverter: Since $A \odot 0 = \bar{A}$ & $A \odot 1 = A$, we can give an input `A` & use the 2nd input as control. If the control is 1, output is same as input, otherwise output is inverted.

Sum of Product & Canonical Sum of Product

- SoP need not contain all the literals, but in Canonical form, each product term should contain all literals, be it in complemented or un-complemented form.
- The product terms themselves are called the min-terms.
- **Sum of all min-terms for which output $f = 1$, is called Canonical Sum of Product, or disjunctive normal form**.
- Truth Table:

x	y	z	Decimel	f
0	0	0	0	0
0	0	1	1	1
0	1	0	2	0
0	1	1	3	1
1	0	0	4	0
1	0	1	5	1
1	1	0	6	0
1	1	1	7	1

  - If $x = 0$, we write $\bar{x}$, otherwise we write x.
  - So, SoP ie $f(1) = \bar{x}\bar{y}z + \bar{x}yz + x\bar{y}z + xyz$
  - We can also write it as $\sum m(1, 3, 5, 7)$ or $\sum(m_1 + m_3 + m_5 + m_7)$.

Duality Theorem

To get the dual of any boolean expression, replace:

Source	Destination
OR	AND
.	+

Source	Destination
NOT	*keep as-is*
XOR	XNOR
NAND	NOR
0	1
Variable	*keep as-is*

- Formulae

- **Complement**: Has all properties of the Duality Theorem, and we complement the variables in addition.
  - $A \ <-> \ \bar{A}$
  - $\bar{A} \ <-> \ A$
- Example 0: $XOR = \bar{A}B + A\bar{B}$
  - $(\bar{A} + B).(A + \bar{B})$
  - $\bar{A}A + AB + \bar{A}\bar{B} + B\bar{B}$
  - $(AB + \bar{A}\bar{B}) = XNOR$ [Duality]
  - $(\bar{A}\bar{B} + AB)$ [Complement]
- Example 1: $(AB\bar{C}) + (\bar{A}BC) + (ABC)$
  - $(A + B + \bar{C}).(\bar{A} + B + C).(A + B + C)$ [Duality]
  - $(\bar{A} + \bar{B} + C).(A + \bar{B} + \bar{C}).(\bar{A} + \bar{B} + \bar{C})$ [Complement]
- Example 2: $(XYZ) + (\bar{X}Y\bar{Z}) + (\bar{Y}Z) = 1$
  - $(X + Y + Z).(\bar{X} + Y + \bar{Z}).(\bar{Y} + Z) = 0$ [Duality]
  - $(\bar{X} + \bar{Y} + \bar{Z}).(X + \bar{Y} + Z).(Y + \bar{Z}) = 0$ [Complement]
- Example 3, demonstrating **self-dual equation**: $XY + YZ + XZ)$
  - $(X + Y).(Y + Z).(X + Z)$
  - $(XY + XZ + Y + YZ).(X + Z)$ # `Y.Y` can be written as `Y`
  - $(Y[X + 1 + YZ] + XZ).(X + Z)$
  - $(Y + XZ)(X + Z)$ # 1 added with anything will result in `1`, so `X+1+YZ` is resolved to `1`.

- $XY + YZ + XZ + XZ$ # `XXZ=XZ` & `XZZ=XZ`
- $XY + YZ + XZ$ # `XZ+XZ=XZ`
- This kind of equation is called a self-dual equation. In other words, the output is the same as input in such an equation.

* Example 4: How many self-dual functions can be made with 1 variable?
  - Total number of combinations: 2, ie A can be either 0 or 1
  - Total number of Boolean functions: 4, ie $0, 1, A$ or $\bar{A}$
  - Total number of self-dual functions: 2 (out of $0, A, \bar{A}, 1$, only $A, \bar{A}$ are self-dual)

A	f_1	f_2	f_3	f_4
0	0	0	1	1
1	0	1	0	1
Result	0	A	$\bar{A}$	1

* Example 5: How many self-dual functions can be made with 2 variables?
  - Total number of combinations: **4**, ie A can be either 0 or 1 & B can also be either 0 or 1
  - Total number of Boolean functions: **16**.
  - Total number of self-dual functions: **4**, ie only $A, \bar{A}, B, \bar{B}$ are self-dual out of all functions.

A	B	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}
0	0	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0
1	0	0	0	1	1	0	0	1	1	0	0	1
1	1	0	1	0	1	0	1	0	1	0	1	0

K-Map

- aka **Karnaugh Map**

- A K-Map is used to **graphically represent & minimize** boolean expressions.
- For a boolean expression of `n` variables, number of cells needed in K-Map is 2^n.
- K-Map is based on Grey code (unit distance code). We can't change more than 1 bit in a single step.
- **Prime Implicants**: Min-terms which have a `1` in them.
- **Essential Prime Implicants**: Min-terms which have a `1` in them, which is also not shared with other pairs.
- There are 3 types of input values, `0`, `1`, `d,x (don't care)`.
- Steps:
    i. We generate the K-Map.
    ii. We find the pairs. They contain 1 mandatorily. We may take `d` ie `don't care` if needed, otherwise we ignore them. While pairing elements, we first try to find the biggest pair possible (16 elements in a K-Map of 16 elements). Then, we gradually decrease the pair size.
    iii. We find the min-terms. Min-terms consist of variables which are same/common for all elements of the pair.
- Example: Generate a K-Map for (A, B, C) & (A, B, C, D)
    - Detailed versions (the cell values denote the decimel representation of the positions):
        - 2 variables, AB:

$\downarrow A \mid B \rightarrow$	$\bar{B}_{(0)}$	$B_{(1)}$
$\bar{A}_{(0)}$	$0_{(00)}$	$1_{(01)}$
$A_{(1)}$	$2_{(10)}$	$3_{(11)}$

        - 3 variables, ABC:

$\downarrow A \mid BC \rightarrow$	$\bar{B}\bar{C}_{(00)}$	$\bar{B}C_{(01)}$	$BC_{(11)}$	$B\bar{C}_{(10)}$
$\bar{A}_{(0)}$	$0_{(000)}$	$1_{(001)}$	$3_{(011)}$	$2_{(010)}$

$\downarrow A \mid BC \rightarrow$	$\bar{B}\bar{C}_{(00)}$	$\bar{B}C_{(01)}$	$BC_{(11)}$	$B\bar{C}_{(10)}$
$A_{(1)}$	$4_{(100)}$	$5_{(101)}$	$7_{(111)}$	$6_{(110)}$

- 4 variables, $ABCD$:

$\downarrow AB \mid CD \rightarrow$	$\bar{C}\bar{D}_{(00)}$	$\bar{C}D_{(01)}$	$CD_{(11)}$	$C\bar{D}_{(10)}$
$\bar{A}\bar{B}_{(00)}$	$0_{(0000)}$	$1_{(0001)}$	$3_{(0011)}$	$2_{(0010)}$
$\bar{A}B_{(01)}$	$4_{(0100)}$	$5_{(0101)}$	$7_{(0111)}$	$6_{(0110)}$
$AB_{(11)}$	$12_{(1100)}$	$13_{(1101)}$	$15_{(1111)}$	$14_{(1110)}$
$A\bar{B}_{(10)}$	$8_{(1000)}$	$9_{(1001)}$	$11_{(1011)}$	$10_{(1010)}$

- Simplified versions & templates:

  - 2 variables, AB:

$\downarrow A \mid B \rightarrow$	$\bar{B}$	B
$\bar{A}$	$null_{(0)}$	$null_{(1)}$
A	$null_{(2)}$	$null_{(3)}$

  - 3 variables, ABC:

$\downarrow A \mid BC \rightarrow$	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	$null_{(0)}$	$null_{(1)}$	$null_{(3)}$	$null_{(2)}$
A	$null_{(4)}$	$null_{(5)}$	$null_{(7)}$	$null_{(6)}$

  - 4 variables, $ABCD$:

$\downarrow AB \mid CD \rightarrow$	$\bar{C}\bar{D}_{(00)}$	$\bar{C}D_{(01)}$	$CD_{(11)}$	$C\bar{D}_{(10)}$
$\bar{A}\bar{B}_{(00)}$	$null_{(0)}$	$null_{(1)}$	$null_{(3)}$	$null_{(2)}$
$\bar{A}B_{(01)}$	$null_{(4)}$	$null_{(5)}$	$null_{(7)}$	$null_{(6)}$
$AB_{(11)}$	$null_{(12)}$	$null_{(13)}$	$null_{(15)}$	$null_{(14)}$

$\downarrow AB \mid CD \rightarrow$	$\bar{C}\bar{D}_{(00)}$	$\bar{C}D_{(01)}$	$CD_{(11)}$	$C\bar{D}_{(10)}$
$A\bar{B}_{(10)}$	$null_{(8)}$	$null_{(9)}$	$null_{(11)}$	$null_{(10)}$

- Example 1: $f(A, B) = \sum(2, 3)$
  - Method 1, by directly solving the equation:

A	**B**	f
0	0	0
0	1	0
1	0	1
1	1	1

    - As per the question, $f = 1$ for index 2 & 3 (within 0-3).
    - The resultant equation is: $A\bar{B} + AB = A(\bar{B} + B) = A.1 = A$
  - Method 2, using K-Map:

null	$\bar{B}$	B
$\bar{A}$	$0_{(0)}$	$0_{(1)}$
A	$1_{(2)}$	$1_{(3)}$

    - Pair: (2) & (3). Output: A
- Example 2: $\sum m(0, 2, 5, 7, 9, 11) + d(3, 8, 10, 12, 14)$

$\downarrow PQ \mid RS \rightarrow$	$\bar{R}\bar{S}_{(00)}$	$\bar{R}S_{(01)}$	$RS_{(11)}$	$R\bar{S}_{(10)}$
$\bar{P}\bar{Q}_{(00)}$	$1_{(0)}$	$null_{(1)}$	$d_{(3)}$	$1_{(2)}$
$\bar{P}Q_{(01)}$	$null_{(4)}$	$1_{(5)}$	$1_{(7)}$	$null_{(6)}$
$PQ_{(11)}$	$d_{(12)}$	$null_{(13)}$	$null_{(15)}$	$d_{(14)}$
$P\bar{Q}_{(10)}$	$d_{(8)}$	$1_{(9)}$	$1_{(11)}$	$d_{(10)}$

  - Pairs: $\{8, 9, 11, 10\}, \{0, 2, 8, 10\}, \{5, 7\}$
  - Min-Terms: **4** ie $P\bar{Q}, \bar{Q}\bar{S}, \bar{P}QS, \bar{P}\bar{Q}\bar{S}$

- Essential Prime Implicants: **3** ie $P\bar{Q}, \bar{Q}\bar{S}, \bar{P}QS$
- Example 3: $\sum m(5, 11, 13, 14, 15)$

$\downarrow AB \mid CD \rightarrow$	$\bar{C}\bar{D}_{(00)}$	$\bar{C}D_{(01)}$	$CD_{(11)}$	$C\bar{D}_{(10)}$
$\bar{A}\bar{B}_{(00)}$	$null_{(0)}$	$null_{(1)}$	$null_{(3)}$	$null_{(2)}$
$\bar{A}B_{(01)}$	$null_{(4)}$	$1_{(5)}$	$null_{(7)}$	$null_{(6)}$
$AB_{(11)}$	$null_{(12)}$	$1_{(13)}$	$1_{(15)}$	$1_{(14)}$
$A\bar{B}_{(10)}$	$null_{(8)}$	$null_{(9)}$	$1_{(11)}$	$null_{(10)}$

  - Pairs: $\{5, 13\}, \{13, 15\}, \{15, 14\}, \{15, 11\}$
  - Prime Implicants: **4** ie $B\bar{C}D, ABD, ABC, ACD$
  - Essential Prime Implicants: **3** ie $B\bar{C}D, ABC, ACD$

Digital Logic Circuits

Half-Adder

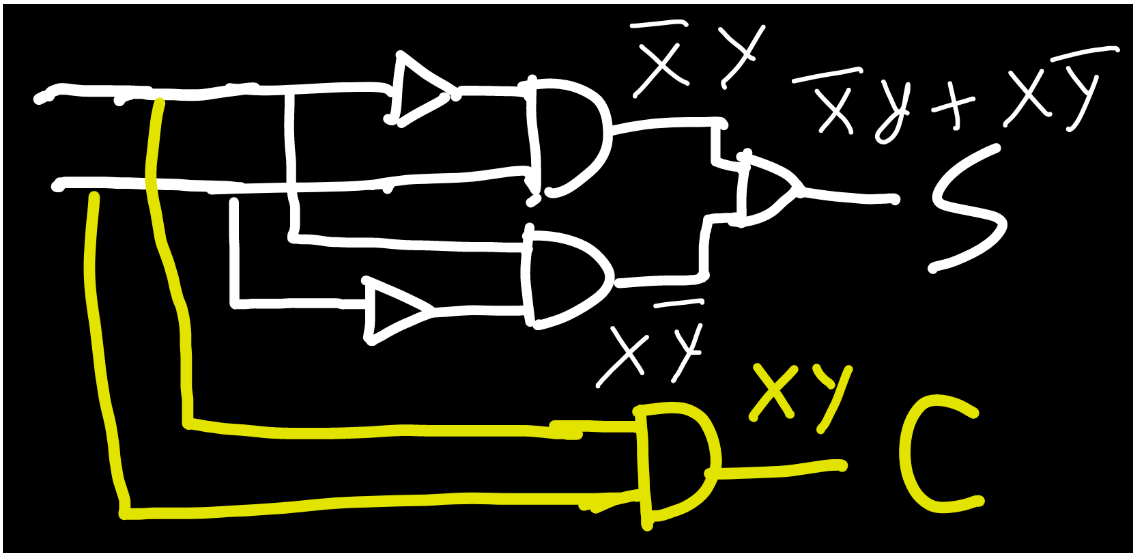
- Adds 2 bits
- Inputs: 2 | Outputs: 2
- Sum (Least Significant Bit [LSB]): $x \oplus y$
- Carry (Most Significant Bit [MSB]): xy
- Truth Table:

X	Y	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

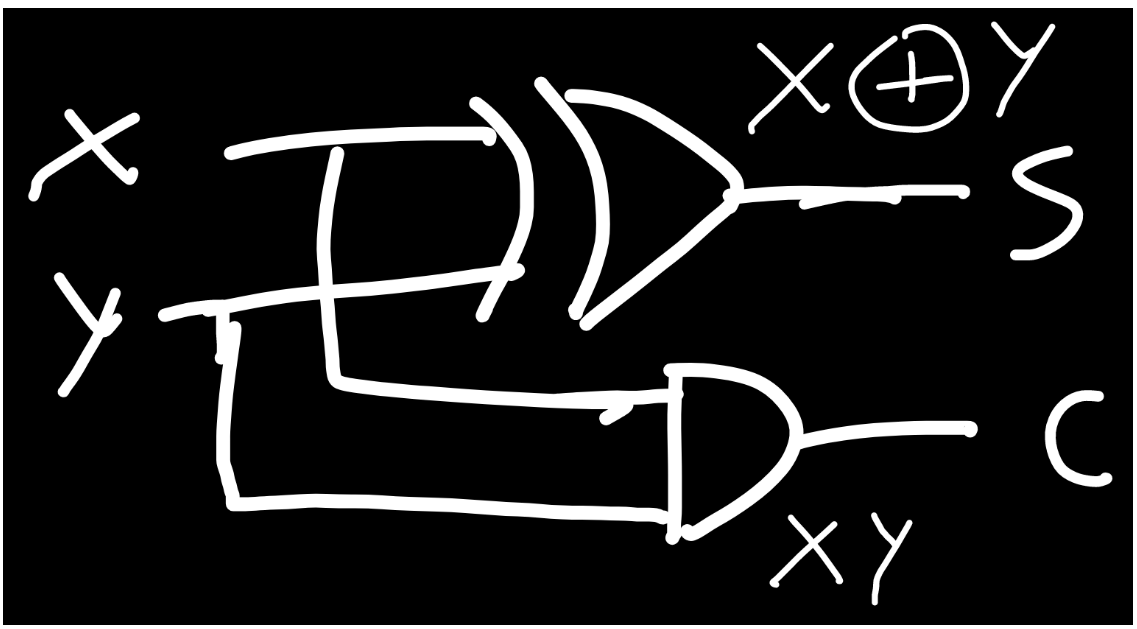
- Min-terms:
  - Sum: $\bar{x}y, x\bar{y} = x \oplus y$

- Carry: xy

- Circuit Diagram:
  - Sum: $\bar{x}y, x\bar{y}$ | Carry: xy

  - Sum: $x \oplus y$ | Carry: xy

Full-Adder

- Adds 3 bits
- Inputs: 3 | Outputs: 2
- 2 Half-Adders = Full-Adder
- Sum: $x \oplus y \oplus z$
- Carry: $xy + yz + zx = (x \oplus y)z + xy$

- Truth Table:

x	y	z (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

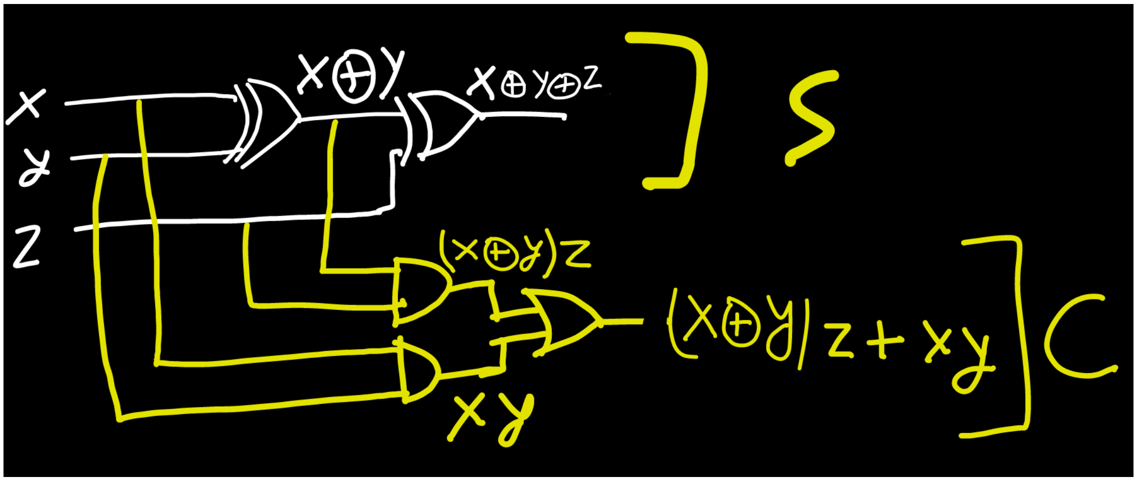
- Min-Terms:
  - Sum: $\sum(m_1, m_2, m_4, m_7) = x \oplus y \oplus z$

$\downarrow A \mid BC \rightarrow$	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	$null_{(0)}$	$1_{(1)}$	$null_{(3)}$	$1_{(2)}$
A	$1_{(4)}$	$null_{(5)}$	$1_{(7)}$	$null_{(6)}$

    - No pairs possible
  - Carry: $\sum(m_3, m_5, m_6, m_7) = xy + yz + xz = (x \oplus y)z + xy$

$\downarrow A \mid BC \rightarrow$	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	$null_{(0)}$	$null_{(1)}$	$1_{(3)}$	$null_{(2)}$
A	$null_{(4)}$	$1_{(5)}$	$1_{(7)}$	$1_{(6)}$

    - Pairs: $BC, AC, AB = xy + yz + xz$
- Circuit Diagram:
  - Sum: $x \oplus y \oplus z$ | Carry: $(x \oplus y)z + xy$


x
y
z
x⊕y
x⊕y⊕z
S
(x⊕y)z
(x⊕y)z + xy
xy
C