ESD - Elettronica dei Sistemi Digitali

Solutions on Graphical Minimization Methods

Prof. Riccardo Berta

2025.11.03

1 Graphical minimization methods Exercises

1.1 Exercise 1

Design the following circuits by proceeding through these steps: define the truth table based on the textual description, construct the corresponding Karnaugh map, minimize the Boolean function, draw the circuit schematic, and finally verify the correctness of the truth table using Deeds.

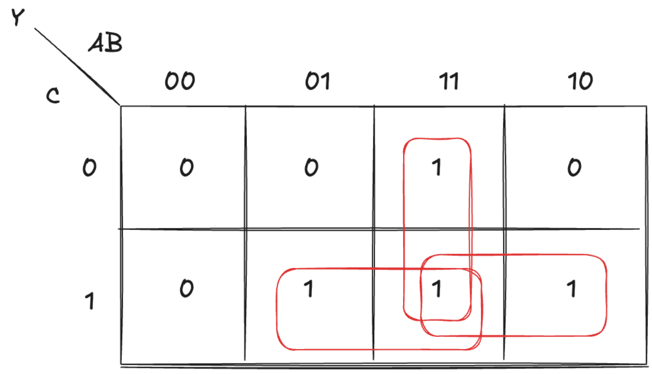
1.1.1 (a) Majority Detector

Design a circuit with three inputs A, B, C and one output Y, which is 1 whenever at least two inputs are 1.

Truth Table:

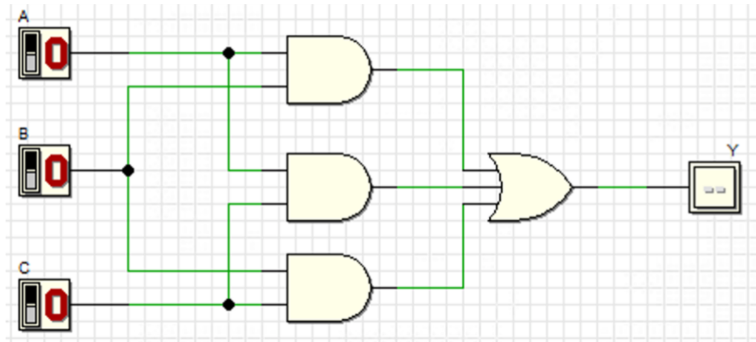
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

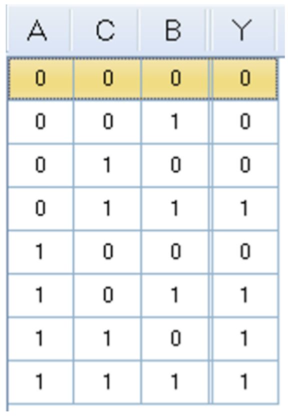
Karnaugh Map:



Minimized Boolean Function: $Y = AB + AC + BC$

Circuit Schematic and verification:





A	C	B	Y
0	0	0	0
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	1

1.1.2 (b) Odd Parity Detector

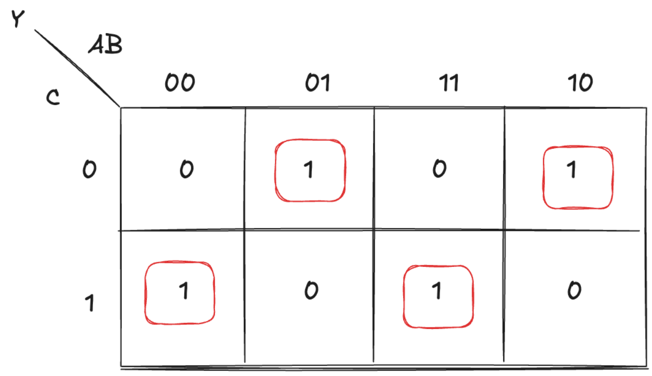
Design a circuit with three inputs A, B, C and output Y = 1 if the number of 1s is odd

Truth Table:

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

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

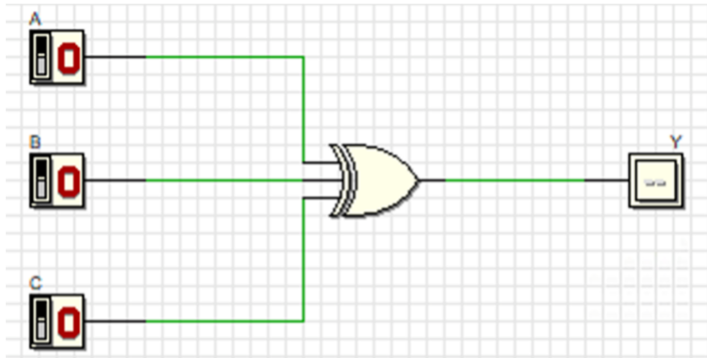
Karnaugh Map:

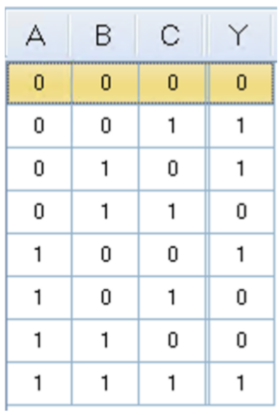


The Karnaugh maps for the parity checker clearly show a **symmetry** called **checkerboard pattern**. This pattern reflects the nature of the function: the output is 1 whenever the number of 1s among the inputs is odd. This pattern can be exploited to directly write the minimized Boolean function as an XOR operation among all the inputs:

$Y = A \oplus B \oplus C$

Schematic and verification:





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

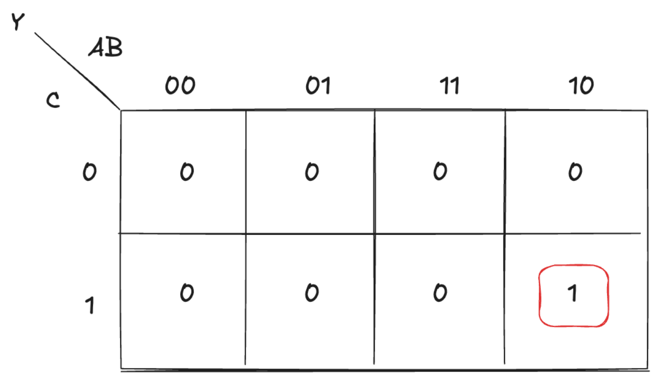
1.1.3 (c) Code Detector

Design a circuit with three inputs A, B, C. The output Y = 1 only when the input pattern is 101.

Truth Table:

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

Karnaugh Map:

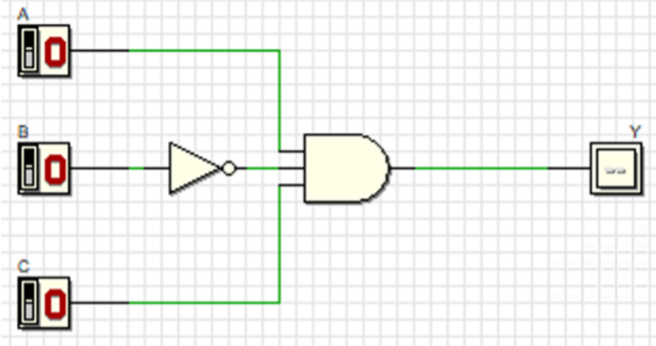


The function corresponds to a single minterm:

$Y = A\overline{B}C$

This circuit can be implemented with one AND gate taking inputs. It represents the simplest form of a combinational detector, producing a high output only for an exact binary pattern 101. For example, it can be used in digital systems to recognize specific command codes or data sequences.

Schematic and verification:



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

1.1.4 (d) Greater Than 9

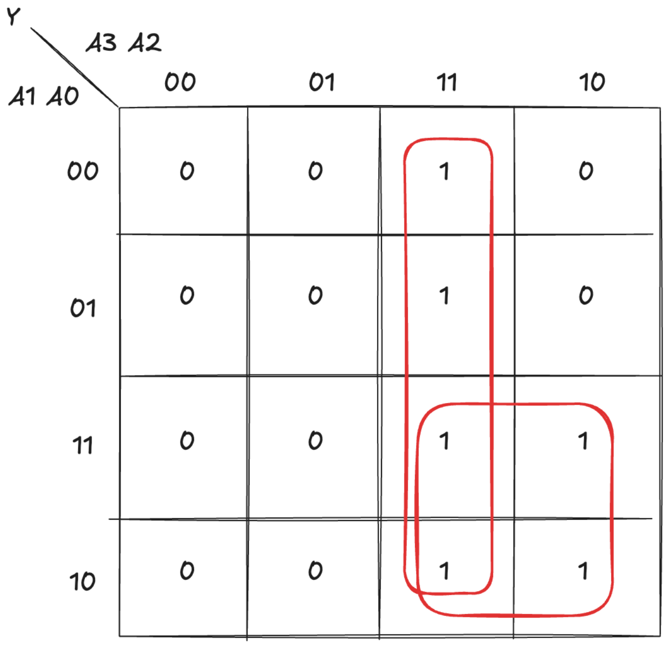
Design a circuit that takes a 4-bit input A3 A2 A1 A0 and outputs Y=1 if the unsigned decimal number represented by the binary input is greater than 9 (decimal 10–15).

Trueth Table:

A3	A2	A1	A0	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1

A3	A2	A1	A0	Y
1	1	1	0	1
1	1	1	1	1

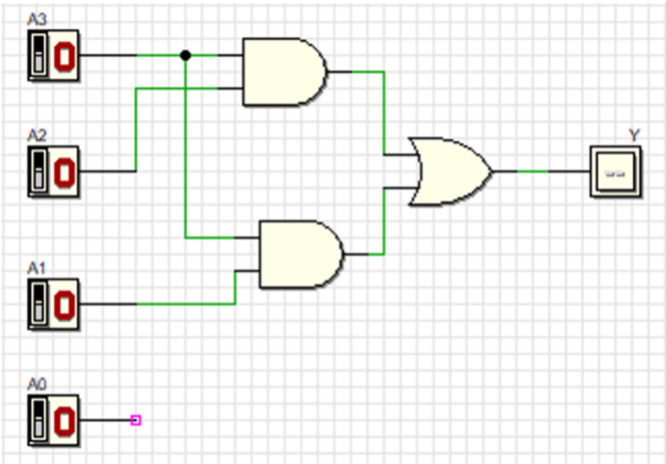
Karnaugh Map:



$Y = A_3A_2 + A_3A_1$

In this circuit, the variable A0 (the least significant bit) does not affect the output, since the condition ″number > 9″ depends only on the three most significant bits (A3, A2 and A_1). In other words, A0 is **irrelevant (a don't-care variable)** for this function, which is why it does not appear in the simplified Boolean expression.

Schematic and verification:



A3	A2	A1	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

1.1.5 (e) 2-bit Comparator

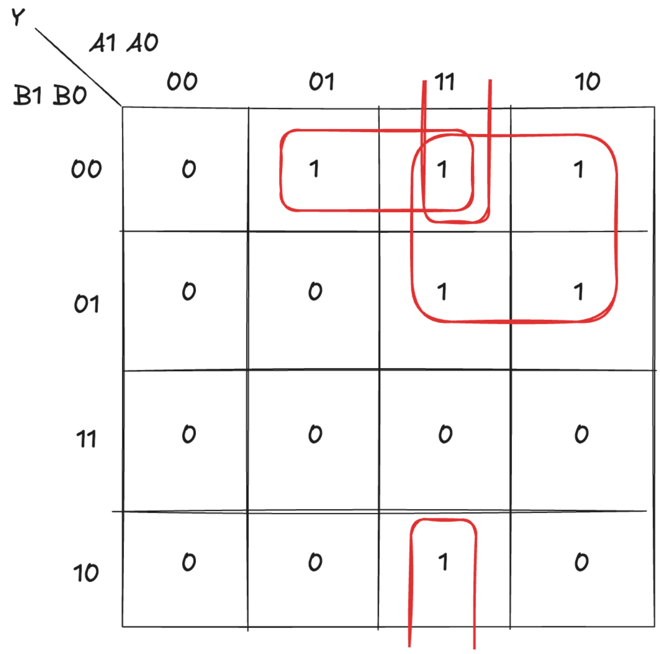
Design a circuit with two 2-bit binary numbers A1 A0 and B1 B0. The output Y = 1 if the unsigned decimal number represented by A is greater than the unsigned decimal number represented by B.

Truth Table:

A1	A0	B1	B0	A (decimal)	B (decimal)	Y
0	0	0	0	0	0	0
0	0	0	1	0	1	0
0	0	1	0	0	2	0
0	0	1	1	0	3	0
0	1	0	0	1	0	1
0	1	0	1	1	1	0
0	1	1	0	1	2	0
0	1	1	1	1	3	0
1	0	0	0	2	0	1
1	0	0	1	2	1	1
1	0	1	0	2	2	0
1	0	1	1	2	3	0

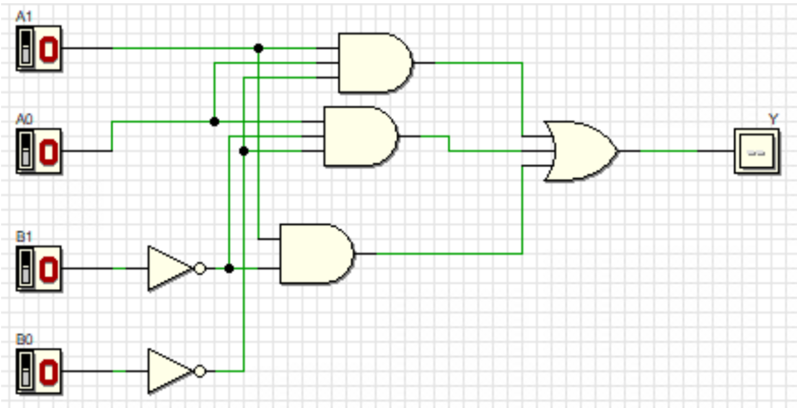
A1	A0	B1	B0	A (decimal)	B (decimal)	Y
1	1	0	0	3	0	1
1	1	0	1	3	1	1
1	1	1	0	3	2	1
1	1	1	1	3	3	0

Karnaugh Map:



$Y = A_1\overline{B_1} + A_0\overline{B_0}\,\overline{B_1} + A_1 A_0\overline{B_0}$

Schematic and verification:



A1	A0	B1	B0	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

1.1.6 (f) Binary-to-Gray Code Converter

Design a circuit that converts a 3-bit binary input (A2 A1 A0) to its Gray code (G2 G1 G0)

Truth Table:

A2	A1	A0	G2	G1	G0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

Karnaugh Map:

G2 — A0 \ A2 A1	00	01	11	10
0	0	0	1	1
1	0	0	1	1

$G_2 = A_2$

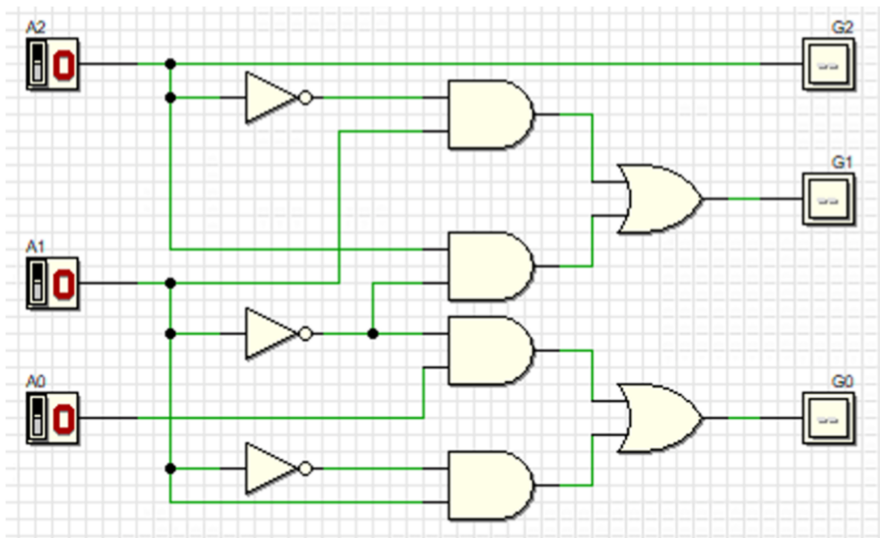
G1 — A0 \ A2 A1	00	01	11	10
0	0	1	0	1
1	0	1	0	1

$G_1 = A_2\overline{A_1} + \overline{A_2}A_1$

G0 — A0 \ A2 A1	00	01	11	10
0	0	1	1	0
1	1	0	0	1

$G_0 = A_1\overline{A_0} + \overline{A_1}A_0$

Schematic and verification:



A2	G2
0	0
1	1

A2	A1	G1
0	0	0
0	1	1
1	0	1
1	1	0

A1	A0	G0
0	0	0
0	1	1
1	0	1
1	1	0

1.2 Exercise 2

Simplify the following Boolean expressions using two different approaches: using Boolean algebra theorems and Karnaugh maps.

1.2.1 (a)

$G = ABC + B\overline{C}$

Boolean Algebra:

$G = B(AC + \overline{C})$

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

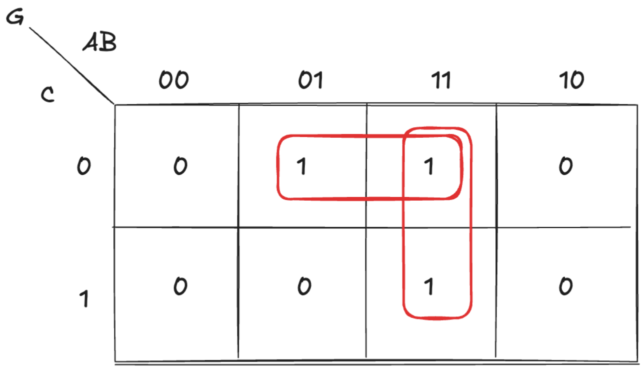
$G = AB + B\overline{C}$

Truth Table:

A	B	C	G
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0

A	B	C	G
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Karnaugh Map:



$G = AB + B\overline{C}$

1.2.2 (b)

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

Boolean Algebra:

$H = AB + A\overline{C} + \overline{B}B + \overline{B}\,\overline{C}$

$H = AB + A\overline{C} + \overline{B}\,\overline{C}$

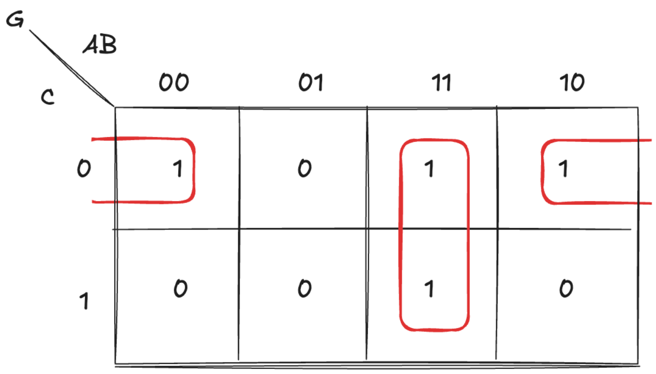
$H = AB + \overline{B}\,\overline{C}$

Truth Table:

A	B	C	H
0	0	0	1
0	0	1	0
0	1	0	0

A	B	C	H
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Karnaugh Map:



$H = AB + \overline{B}\,\overline{C}$

1.2.3 (c)

$Y = BC\overline{D} + CD + A\overline{B}C\overline{D} + \overline{A}\,\overline{B}C$

Boolean Algebra:

$Y = BC\overline{D} + CD + A\overline{B}C\overline{D} + \overline{A}\,\overline{B}C$

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

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

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

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

$Y = CB + CD + \overline{B}C(\overline{D} + \overline{A})$

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

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

$Y = CD + C\overline{D} + C\overline{A} + CB$

$Y = C(D + \overline{D}) + C\overline{A} + CB$

$Y = C + C\overline{A} + CB$

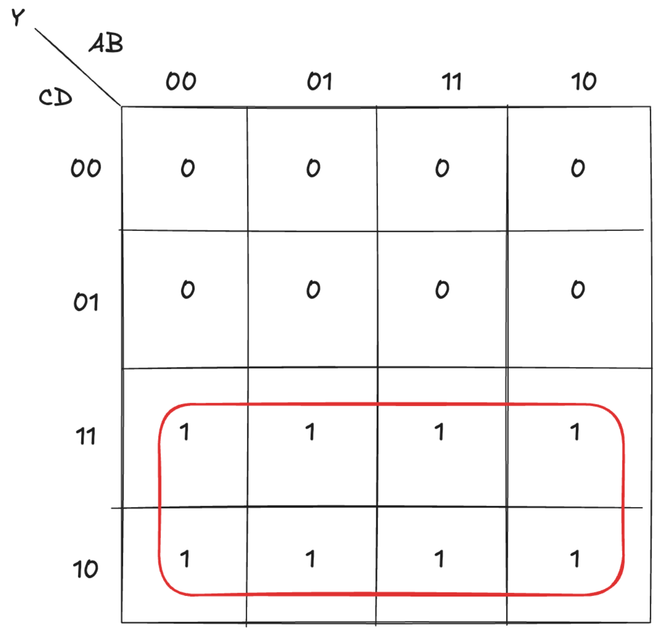
$Y = C(1 + \overline{A} + B)$

$Y = C$

Truth Table:

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

Karnaugh Map:



$Y = C$

1.2.4 (d)

$Y = AB + A\overline{B}\,\overline{C}\,\overline{D}$

Boolean Algebra:

$Y = AB + A\overline{B}\,\overline{C}\,\overline{D}$

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

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

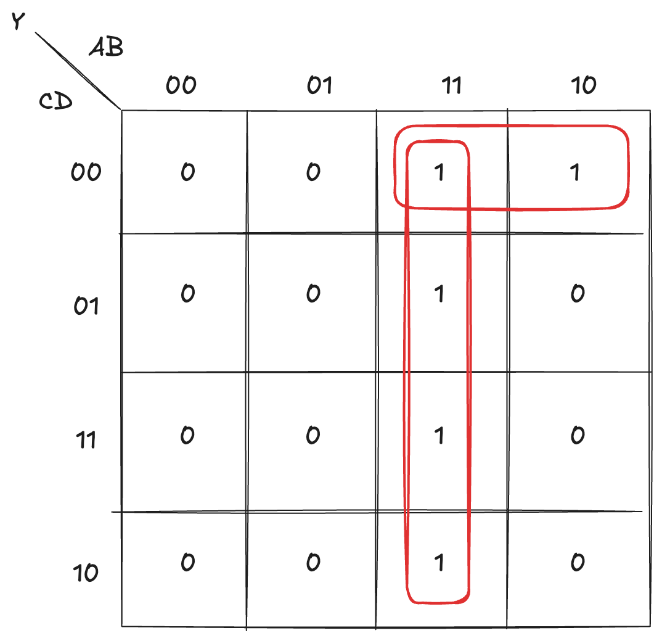
$Y = AB + A\overline{C}\,\overline{D}$

Truth Table:

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0

A	B	C	D	Y
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Karnaugh Map:



$Y = AB + A\overline{C}\,\overline{D}$