Karnaugh maps

- PRO's of smaller number of gates:
  - minimize use of different type of gates
  - lower cost of product
    - ↳ simpler gates is easier to produce = less mistakes made

- Logic gates are quick yet use low energy
  - ↳ don't get overworked
  - ↳ lessen prescribed number of I/O ports needed by microcontroller
  - ↳ straightforward data encryption & decryption

- Generic approach for minimising Boolean functions
  = Karnaugh maps (K-maps)
  - ↳ use graphical representation to find cases
  - ↳ where different terms in Boolean formula
  - ↳ can be combined to <u>one simpler term</u>
  - ↳ eg. **FROM:** $(A \wedge B) \vee (A \wedge \bar{B}) = A \wedge (B \vee \bar{B}) = A \wedge 1 = A$

    TO: $(A \wedge B \wedge C) \vee (A \wedge \bar{B} \wedge C) = A \wedge C$

- EXAMPLES:

  ↳ $F(A,B) = \boxed{AB} + A\bar{B}$ → $A \wedge B$; → $A + B \Rightarrow A \vee B$

  ↳

A \ B	0	1
0	0	0
1	1	1

→ each group of 1 that's not diagonal represents a group of terms that can be combined into one term

↳ $A = 1$

so

$F(A,B) = AB + A\bar{B} = A$

K-map with 3 variables :

Fantasy

$F(A,B,C) = AB\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + ABC + \overline{AB}C$

↳ 5 terms :

valid for $A = 1$

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

4, is a large group

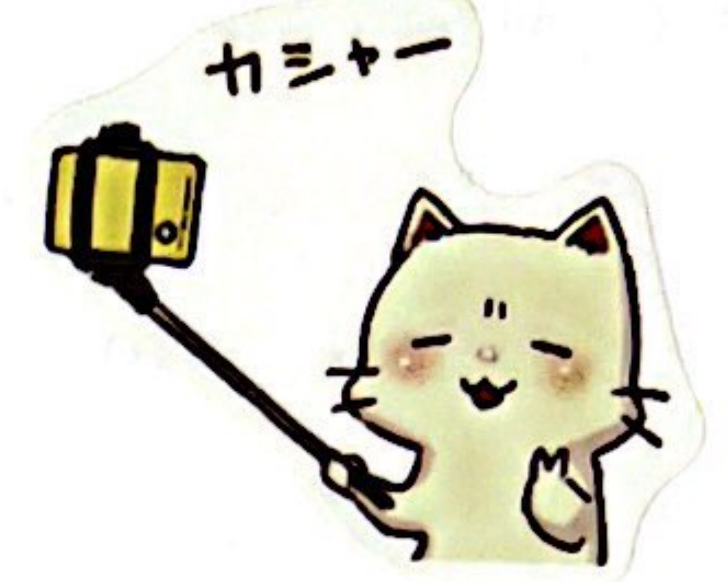
wrapping round to the other side for another group

valid for $C=1$, $B=0$; independant from A

$\therefore F(A,B,C) = A + \bar{B}C$

- rules for K-maps

#1 = no group can contain 0



A \ B	0	1
0	0	1
1	0	1

CORRECT

A \ B	0	1
0	0	1
1	0	1

WRONG

#2 : groups can be horizontal, vertical, square but NOT diagonal

A \ B	0	1
0	0	1
1	1	1

CORRECT

A \ B	0	1
0	0	1
1	1	1

WRONG

#3 : groups must contain 1, 2, 4, 8, 16, 32 ... (power of 2)

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

CORRECT

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

WRONG, 3 is not power of 2

#4 : Each group must be as large as possible

#5 : groups can overlap

#6 : Each 1 must be part of at least 1 group

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

correct ✓ as large as possible
✓ overlap ✓ power of 2

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

WRONG! as large as possible!

#7 : groups may wrap around the map

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

✓ wrap around
✓ power of 2