Today

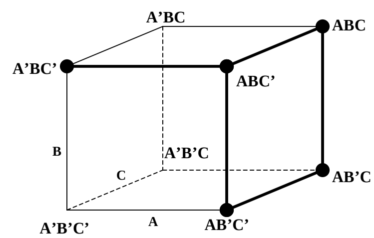
- K-maps
- Reverse Engineering Example
- Max block example

Karnaugh Maps

- What was the idea in doing simplication? Well, one of the ideas was to try to apply the unification theorem (AB + AB′ = A).
- What we're looking for then are terms that differ only in one variable.
- This can be difficult to do when there are many terms and many variables. K-maps are just a graphical method that makes it easier.

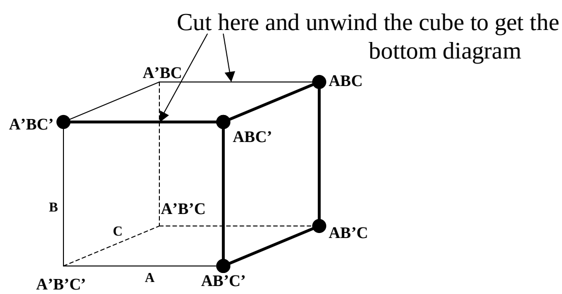
Cube Representation.

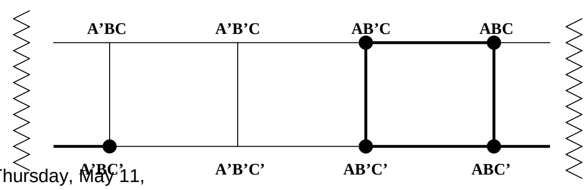
- Draw a cube: each vertex is a possible term, AND two adjacent vertices only differ in one variable.
- Now, draw a dot for each term from our boolean expression, and group dots that are connected.
- An edge that connects two dots means that we can apply the unification theorem to merge those two terms. The variable that differs is dropped.
- By applying the unification theorem twice, we can merge 4 vertices that are fully connected.



The above cube shows the expression A'BC + ABC' + ABC + AB'C + AB'C'.
It simplifies to:
A + BC'

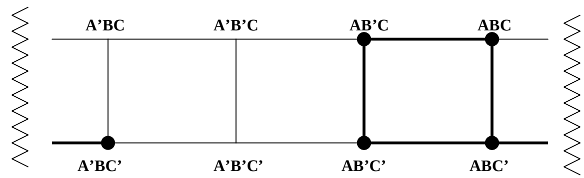
Karnaugh Maps (cont'd)





Note how, because we are now in 2d, the lines kinda wrap around.

Karnaugh Maps (cont'd)



A simpler drawing of this is given below. We call this kind of table a Karnaugh map.

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

Implicants

- Implicant: A product term whose "oneness" *implies* the functions "oneness".
- Prime Implicant: Implicant that cannot be combined with another implicant.
- Essential Prime Implicant: Implicant that covers an element of the on-set which is not covered by any other implicant.

Example of Implicants

- Implicants
- Six Prime Implicants: A’B’D, BC’, AC, A’C’D, AB, B’CD
- Essential PI: AC,BC’
- F=A’B’D+BC’+AC

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

K-map: SoP and PoS

- SoP:

A’BC’D’+A’B’C’D+ABC’D+AB’C’D+A’B’CD+AB’CD

- Minimized Exp

A’BC’D’+B’D+AC’D

- PoS:

(A+B+C+D)(A’+B’+C+D)(A’+B+C+D) (A+B’+C+D’) ….

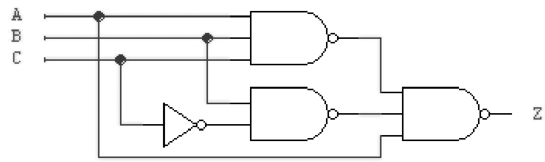
- Minimized PoS

(B+D)(A’+D)(B’+C’)(A+B’+D’)

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

Reverse Engineering Example

- Write down the Boolean expression:



$$f(A,B,C) = \overline{\overline{ABC} \bullet \overline{B\overline{C}} \bullet A}$$

Reverse Engineering Ex (cont.)

- Write the complete truth table for the circuit.

A	B	C	Z
0	0	0	1
	0	1	1
	1	0	1
	1	1	1
1	0	0	0
	0	1	0
	1	0	1
	1	1	1

Reverse Engineering Ex (cont.)

- Do a k-map:

Reverse Engineering Ex (cont.)

- Re-implement the better design.

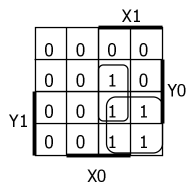
Max function

- The max block has 4 inputs (X1, X0, Y1, Y0), and 2 outputs (M1, M0)
- X1, X0 is a 2 bit 2's complement number, and so is Y1, Y0
- M1, M0 is the maximum between the two input numbers

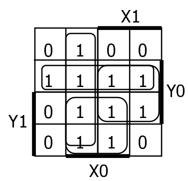
Truth Table

X1	X0	Y1	Y0	M1	M0
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	0	1	0	1
1	0	1	0	1	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

K-map



$M_1 = X_1Y_1 + X_1X_0Y_0$



$M_0 = Y_1'Y_0 + X_1'X_0 + X_1Y_0 + X_0Y_1$