**Optimization of Boolean Functions**

F1 = wxy + wxy’

Diagram

Description automatically generated

There are going to be 18 (4 for or, 2 for not, 6 for per and (there are 3 inputs)) transistors.

This function is same as: wx(y+y’) = wx

Icon

Description automatically generated

Just 4 transistors.

Example:

F = x’y’z’ + x’y’z + x’yz

F = x’y’z’ + x’y’z + x’y’z + x’yz = x’y’(z’ + z) + x’z(y’ + y) = x’y’ + x’z



Example:

F = xy’z’ + xyz + xyz’ + xy’z = xy’(z’ + z) + xy(z + z’) = xy’ + xy = x(y’ + y) = x

Instead of using Boolean algebra, we will use Karnaugh map (K-map) to make simplifications.

Steps:

1. Create your template K-map
2. Fill this K-map using the minterms in the function
3. Combine the minterms for simplification

Assume we have 3 variables (x, y, z). K-map template seem as:

We are finding minterms of the column and rows.

We have placed all of the minterms. Places of these minterms are same. Standard for everybody.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| yz  x | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 | x’y’z’ | x’y’z | x’yz | x’yz’ |
| 1 | xy’z’ | xy’z | xyz | xyz’ |

1. ONLY ONE NUMBER HAS TO BE CHANGED IN CONSECUTIVE ELEMENTS,

THAT’S WHY 1 1 COMES AFTER 0 1 INSTEAD OF 1 0.

1. ROWS EXPECTED TO HAVE LESS NUMBER OF VARIABLES COMPARED TO COLUMNS.

Example:

Represent F in K-map 🡪 F = x’y’z + xyz + xyz’ + x’y’z’

The function that you are going to be represent by K-map should include minterms. It should be in canonical form.

STEP 1-2

STEP 3

For combination, draw a circle to the consecutive (neighbour) 1s (the number of consecutive elements should be even number).

2 consecutive ones. 2 is even so we can draw a circle.

x’y’z’ + x’y’z = x’y’(z’ + z) = x’y’

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| yz  x | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| yz  x | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |



For first circle, blue colored ones are not changing which are x’ and y’. So that means I can express that circle with x’y’. Same as we found with boolean algebra.

For second circle, xy is not changing.

All in all, simplified version of the function F is:

F = xy + x’y’

Example:

F = xy’z’ + xy’z + xyz + xyz’

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| yz  x | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

I HAVE TO MAKE BIGGEST GROUP INCLUDING NEIGHBOUR 1S.

I can also combine them 2 by 2.

x is the only element which not change in all four.

F = x

What happens if you combine them 2 by 2?

xy’ + xy = x(y’ + y) = x 🡪 Same Result

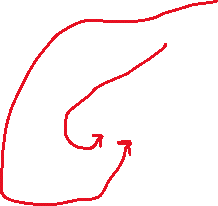


First group Second group

Example:

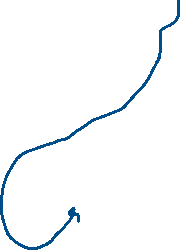
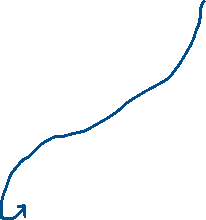
F = x’y’z + xy’z’ + xy’z + xyz + xyz’

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| yz  x | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |



We have different options. Red option:

x + x’y’z



Blue option:

x + y’z

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| yz  x | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 |

3 different combine ways.

Green 🡪 only z is not changing : F = z

A term can be shared by multiple groups.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| yz  x | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

-------------------------OR----------------------------

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| yz  x | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

F = y’z + x

We can regard first and last elements as neighbouring elements. Look at the upper row, first column’s elements is 0 0 and last column’s is 1 0 🡪 only 1 element is changing.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| yz  x | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 |

you can tought these 2 are neighbours

So F = xz’ + x’y’z

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| yz  x | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |

F = x’z + x’y

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| yz  x | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

There is no unchanged element. Doesn’t matter way of the variables x, y, and z, output is 1. If all of the elements are 0, output is going to be 0.

F = 1



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| yz  x | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |



F = x + z’

**IN ALL CASES, YOU HAVE TO MAKE THE BIGGEST CIRCLE.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cd  ab | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 0 | 0 | 0 | 1 | 0 |
| 0 1 | 1 | 1 | 1 | 0 |
| 1 1 | 0 | 0 | 1 | 0 |
| 1 0 | 0 | 0 | 1 | 0 |



F = a’bc’ + cd

STEPS OF K-MAP SIMPLIFICATION:

1. Convert function to sum of minterms
2. place 1s to the appropriate cell in K-map
   * for each minterm, place 1
3. cover all the 1s by creating the largest circle that includes 1s and write the corresponding term
4. OR all the obtained terms

F = a’b’(cd’ + c’d’) + ab’c’d’ + ab’cd’ + a’bd + a’bcd’

a’b’cd’ + a’b’c’d’ + ab’c’d’ + ab’cd’ + a’bd(c’ + c) + a’bcd’

a’b’cd’ + a’b’c’d’ + ab’c’d’ + ab’cd’ + a’bc’d + a’bcd + a’bcd’ 🡺 Canonical Form

4 köşe, 1 grup eder. 🡪 b’d’

Soldaki ikili grup. 🡪 a’bd

Sağdaki ikili grup. 🡪 a’bc

F = b’d’ + a’bd + a’bc

Mavi 1leri de alabilirsin ama uzatırsın, gerek yok. Amaç bütün 1leri alabileceğin bir gruplama yapmak.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cd  ab | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 0 | 1 | 0 | 0 | 1 |
| 0 1 | 0 | 1 | 1 | 1 |
| 1 1 | 0 | 0 | 0 | 0 |
| 1 0 | 1 | 0 | 0 | 1 |

Don’t care combinations:

In some of the cases, you may be sure that some of the combinations will never happen. For example some of the minterms will never appear. It doesn’t matter if they are 1 or 0, we can behave them as 1.

🡪 place x for “don’t care” terms

🡪 don’t have to cover all x, but we can cover any x if we need it

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| yz  x | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 | x | 0 | 0 | 0 |
| 1 | 1 | x | 0 | 0 |

We know that, in some cases 000 and 001 combinations are not going to happen. I can regard them as they are 1 or 0.

I will regard upper left x as 1, other one as 0.

I do not need to cover another circle.

F = y’z’



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| yz  x | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 | 0 | 1 | x | 1 |
| 1 | x | 0 | x | 1 |



F = y + x’z



x’leri 0 da alabilirsin, o zaman sağ dikey 1ler ve solda tekli 1 olur 🡪 F = yz’ + x’y’z 🡪 Daha çok eleman

Line chart

Description automatically generated

