Minimization via K-Maps

Row	A B C	F(A,B,C)
0	0 0 0	0
1	0 0 1	0
2	0 1 0	1
3	0 1 1	0
4	1 0 0	0
5	1 0 1	0
6	1 1 0	1
7	1 1 1	0

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

F(A,B,C) = Σ m(2,6)
= A'BC' + ABC'
= BC'(A'+A)
= BC'

Boolean adjacency can be used to minimize functions!

Groupings on K-Maps

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

F(A,B,C) = BC'

Grouping can be read DIRECTLY as "BC" by looking at what is COMMON within the circled group.

Example Groupings on 3-Variable K-Maps

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

F(A,B,C) = A'B'

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

F(A,B,C) = C'

Remember that top, bottom of map are adjacent!!

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

F(A,B,C) = B'

Multiple Groupings

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

Want to cover all '1's with largest possible groupings.

F(A,B,C) = B'C + A'B'

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

Groupings of only a single '1' are ok if larger groupings cannot be found.

F(A,B,C) = AB'C' + A'B

Illegal Groupings

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

Illegal Grouping! Minterms are not boolean adjacent!

A'B'C' , AB'C will NOT reduce to a single product term
A'B'C' + AB'C = B'(A'C'+AC)

Valid groupings will always be a power of 2. (will cover 1, 2, 4, 8, etc minterms).

Groupings on four Variable Maps

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

F(A,B,C,D) = A'B'C + A'CD' + B'CD' + ABCD

Other Groupings

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

F (A,B,C,D) = B'

More than one way to group…..

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

F (A,B,C,D) = B'D + C'D' + CD'

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

Want LARGEST groupings that can cover '1's.

F (A,B,C,D) = B' + D'

Four Corner Grouping on 4-Variable Map

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

4 Corner grouping is valid on four variable map

F (A,B,C,D) = B'D'

Some Definitions

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

AB'C'

BC

B

Implicant: Any single 1 or any group of 1's is called an implicant of F. Any possible grouping of '1's is an **implicant**.

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

AC'

B

Prime Implicant: A covering that cannot be combined with some other covering to eliminate a variable.

Minimum SOPs

The minimum SOP expression consists of some (but not necessarily all) of the prime implicants of a function.

If a SOP expression contains a term which is NOT a prime implicant, then it CANNOT be minimum.

Prime Implicants

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

EACH of the coverings is a PRIME IMPLICANT.

BC' , A'C'D , A'B'D

Minimum SOP will have some or all of these prime implicants. The included prime implicants must cover all of the ONEs.

F(A,B,C,D) = BC' + A'B'D (minimum # of PIs)
= BC' + A'B'D + A'C'D (valid, but not minimum)
≠ A'B'D + A'C'D (both PI's, but all '1's not included!)

Non-Essential vs. Essential Prime Implicants

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

EACH of the coverings is a PRIME IMPLICANT.

BC' , A'C'D , A'B'D

F(A,B,C,D) = BC' + A'B'D (minimum # of PIs)

Prime Implicant A'C'D is a NON-ESSENTIAL prime implicant because its '1's are covered by other PIs . A PI is ESSENTIAL if it covers a MINTERM that cannot be covered by any other PI.

An example with more than one solution

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

A'C'

ACD

A'BD

BCD

EACH of the coverings is a PRIME IMPLICANT.

Recall that a covering is a Prime Implicant if it cannot be combined with another covering to eliminate a variable.

Two Solutions

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

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

EACH solution is equally valid.

F(A,B,C,D) = A'C' + ACD + A'BD

Essential PIs

Non-Essential PIs

F(A,B,C,D) = A'C' + ACD + BCD

Minimal Solution

A minimal SOP will consist of prime implicants.

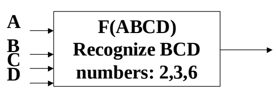
A minimal SOP equation will have all of the essential prime implicants on the map. By definition, these cover a minterm that may not be covered by some other prime implicant.

The minimal SOP equation may or may not include non-essential prime implicants. It will include non-essential prime implicants if there are '1's remaining that have not been covered by an essential prime implicant.

Don't Cares

Row	A B C D	F(A,B,C,D)
0	0 0 0 0	0
1	0 0 0 1	0
2	0 0 1 0	1
3	0 0 1 1	1
4	0 1 0 0	0
5	0 1 0 1	0
6	0 1 1 0	1
7	0 1 1 1	0
8	1 0 0 0	0
9	1 0 0 1	0
10	1 0 1 0	x
11	1 0 1 1	x
12	1 1 0 0	x
13	1 1 0 1	x
14	1 1 1 0	x
15	1 1 1 1	x

Recall that Don't Cares are labeled as 'X's in truth table. Can treat X's as either '0's or '1's



Non BCD numbers are don't cares because will never be applied as inputs.

Don't Cares treated as '0's or '1's

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

Treat X's as 1's when can get a larger grouping. All X's do not have to be covered.

F(A,B,C,D) = CD' + B'C

Minimizing ‘0’s

BC	0	1
00	1	1
01	0	0
11	0	0
10	1	1

Grouping ‘0’s produces an equation for F’.

F(A,B,C) = C’

F’(A,B,C) = C

Minimize 0’s, then Complement to get POS

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

F’(A,B,C,D) = C’ + BD

Take inverse of both sides
F(A,B,C,D) = (C’ + BD)’
= C (BD)’
= C (B’+D’)

Minimizing zeros, then applying inverse to both sides is a way to get to minimum POS form!!!!!

What do you need to know?

- How to minimize functions using 2,3,4 variable Kmaps.
  - Group 1’s to get to minimal SOP form
  - Group 0’s then take complement to get to minimal POS form.
- Definitions of implicant, prime implicant, non-essential prime implicant, essential prime-implicant.
  - Be able to recognize these on a K-map.
- How to treat ‘X’s on a K-map.