CSE 370 Homework 2 Solutions

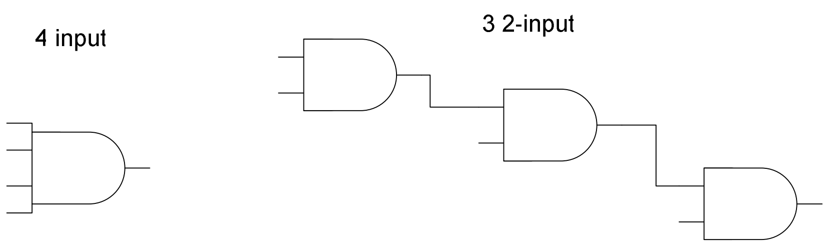
1.

AND: using associativity

$$x_1 \cdot x_2 \cdot x_3 \ldots \cdot x_n = x_1 \cdot (x_2 \cdot (x_3 \cdot \ldots (x_{n-1} \cdot x_n)\ldots))$$

n-1 2-input AND gates can be chained together to form the following circuit that is functionally equivalent to 1 n-input AND gate

<u>example: n=4</u>



NAND: it is impossible to implement a 3-input NAND gate using 2 2-input NAND gates, proved exhaustively

$$((AB)' \cdot C)' = ABC'$$

$$((AB)' \cdot 1)' = AB$$

$$((AB)' \cdot 0)' = 1$$

2. Self Duals

a) self dual

proof: $F = A$

$F^D = A$

b) <u>not</u> self dual

proof: $F = AB' + A'B$

$F^D = (A + B') \cdot (A' + B)$

$F^D = AA' + AB + A'B' + BB'$

$F^D = AB + A'B'$

c) self dual

proof: $F(A,B,C) = \sum(0,3,5,6)$

$F = A'B'C' + A'BC + AB'C + ABC'$

$F^D = (A'+B'+C')(A'+B+C)(A+B'+C)(A+B+C')$

$F^D = (A'+A'B+A'C+A'B'+B'C+A'C'+BC')(A+AB+AC'+AB'+B'C'+AC+BC)$

$F^D = A'B'C' + A'BC + AB'C + ABC'$

d) self dual

proof: for 3 variables

A	B	C	F
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

$F = AB + AC + BC$

$F^D = (A + B)(A + C)(B + C)$

$F^D = (A + AC + AB + BC)(B + C)$

$F^D = (A + BC)(B + C)$

$F^D = AB + AC + BC$

3. Canonical sum and product forms

a) sum:

$F(A,B,C) = A'BC' + AB'C' + ABC' + ABC$

product:

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

b) sum:

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

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

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

product:

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

4. Minimal SOP and POS

a) SOP: $F(A,B,C) = AB + C$

POS: $F(A,B,C) = (B+C)(A+C)$

b) SOP: $F(A,B,C,D) = BC'D' + A'C'D + BCD + ABC$

POS: $F(A,B,C,D) = (A'+B)(B+D)(B+C')(A'+C+D')(A+C'+D)$

5. Cheaper expression for 4b

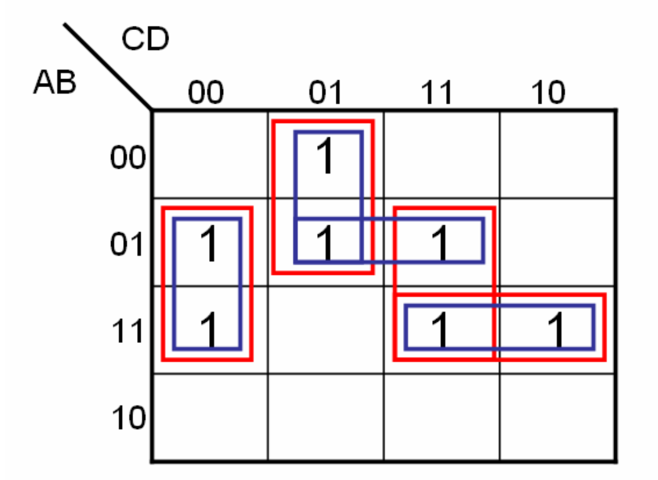
2-level solution in 4b has 14 literals

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

4-level solution here has 12 literals

6. SOP proof

Given the function in 4b as an example, multiple minimal sum-of-products expressions can be chosen by selecting different non-essential prime implicants to implement the function. Example is shown below.



Note: 2 solutions shown here, one in red and one in blue

7. Canonical Functions – problem 2.21 from CLD

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

a) SOP form:

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

F = ABC + ABC' + AB'C' + A'B'C' + ABC' + AB'C'

F = ABC + ABC' + AB'C' + A'B'C'

F(A,B,C) = Σm(0,4,6,7)

b) POS form:

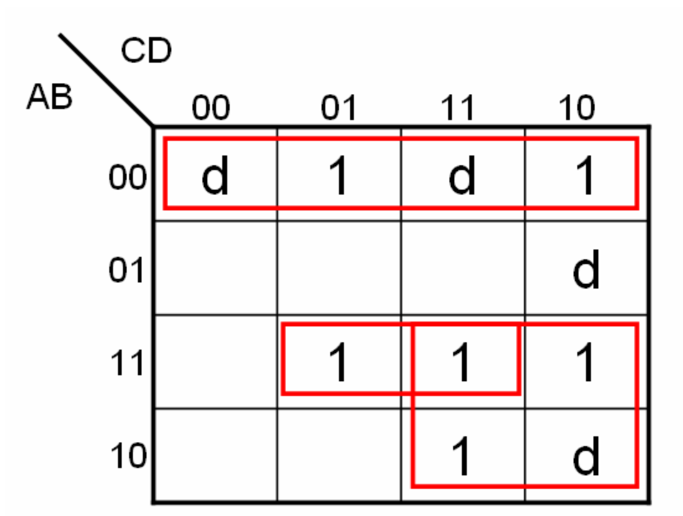
This can be found directly from the SOP form. The SOP form for defining a function targets the 1's of the function, but the POS form targets the zeroes. To find the POS form for this function, simply list the terms that are not covered by the SOP form.

F(A,B,C) = $\prod$M(1,2,3,5)

8. Boolean Simplification – problem 2.26c from CLD

F(A,B,C,D) = Σm(1,2,11,13,14,15) + Σd(0,3,6,10)

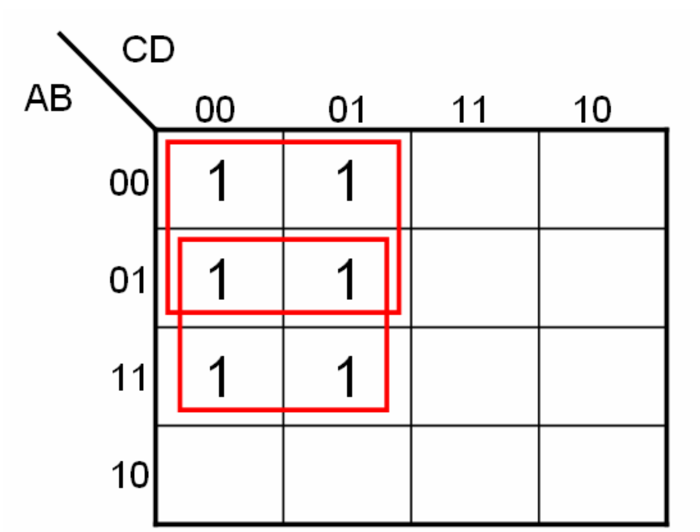
This problem is labeled as Boolean simplification, but it does not say that you must use Boolean algebra to do the simplification. Therefore, the easiest way to solve this problem is through the use of a Karnaugh map, shown below.



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

9. K-map simplification – problem 3.1b from CLD

F(A,B,C,D) = Σm(0,1,4,5,12,13)

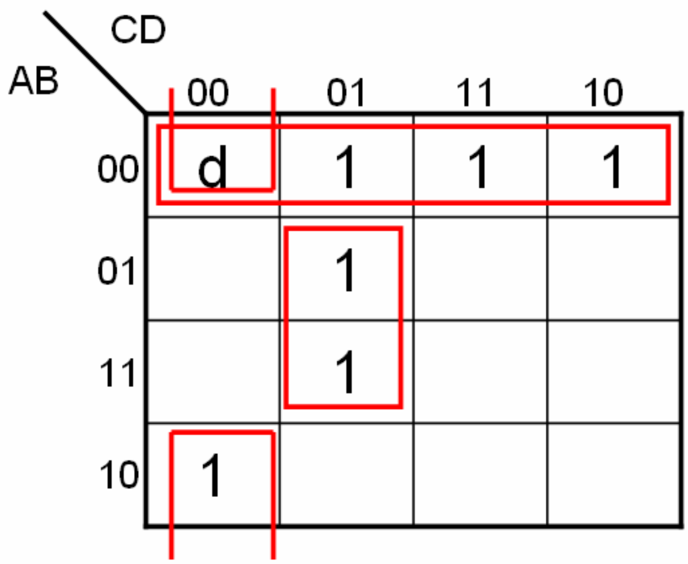


F(A,B,C,D) = A'C' + BC' -this solution contains 4 literals

F(A,B,C,D) = C'(A'+B) -this solution contains 3 literals, but cannot be found only using a k-map

10. K-map simplification and prime implicants – problem 3.3/3.4 c from CLD

$F(A,B,C,D) = \Sigma m(1,2,3,5,8,13) + \Sigma d(0)$



11. General deMorgan's law proof using finite induction

To prove something using finite induction, you prove the base case first. Then you prove the law for the n+1 case, assuming the n^{th} case to be true.

Base case: use perfect induction to prove deMorgan's for n=2

X1	X2	(X1+X2)'	X1'X2'	(X1·X2)'	X1'+X2'
0	0	1	1	1	1
0	1	0	0	1	1
1	0	0	0	1	1
1	1	0	0	0	0

Note: this table proves deMorgan's Law for n=2 through perfect induction

Now, assume deMorgan's law is true for n arguments:

$$(x_1 \cdot x_2 \cdot \ldots \cdot x_n)' = x_1' + x_2' + \ldots + x_n'$$

$$(x_1 + x_2 + \ldots + x_n)' = x_1' \cdot x_2' \cdot \ldots \cdot x_n'$$

Now show for n+1 arguments:

$$(x_1 \cdot x_2 \cdot \ldots \cdot x_n \cdot x_{n+1})' = ((x_1 \cdot x_2 \cdot \ldots \cdot x_n) \cdot x_{n+1})'$$

$$= ((x_1 \cdot x_2 \cdot \ldots \cdot x_n)' + x_{n+1}')$$

$$= x_1' + x_2' + \ldots + x_n' + x_{n+1}'$$

$$(x_1 + x_2 + \ldots + x_n + x_{n+1})' = ((x_1 + x_2 + \ldots + x_n) + x_{n+1})'$$

$$= ((x_1 + x_2 + \ldots + x_n)' \cdot x_{n+1}')$$

$$= x_1' \cdot x_2' \cdot \ldots \cdot x_n' \cdot x_{n+1}'$$