Two Level Logic Optimization : Exact Minimization

Two-Level Logic Minimization

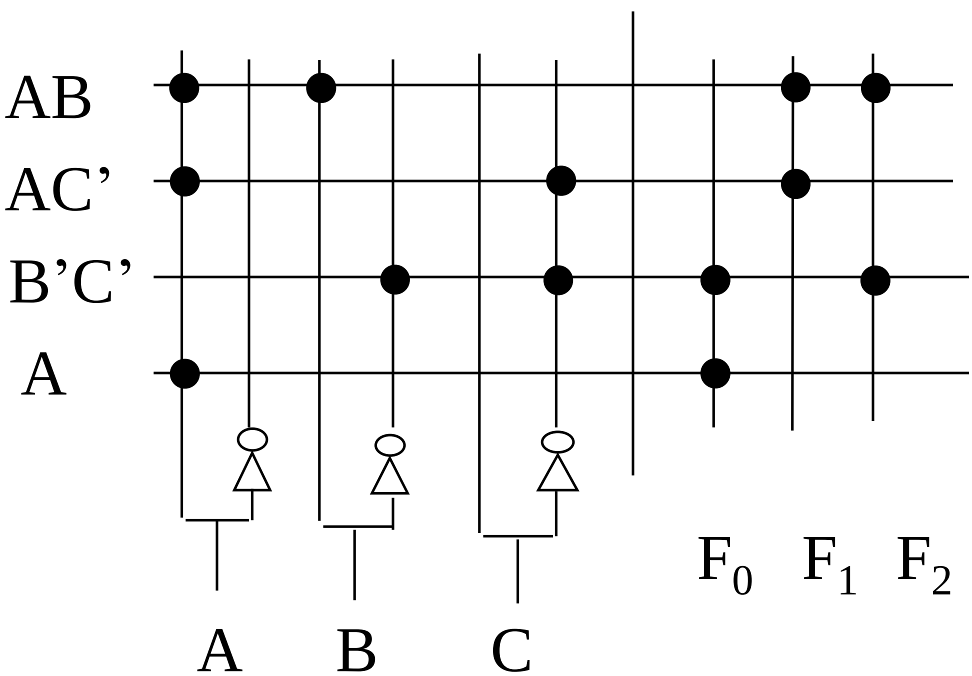
- PLA Implementation

Ex: $F_0 = A + B'C'$

$F_1 = AC' + AB$

$F_2 = B'C' + AB$

product term AB, AC', B'C', A



Goals: 1. product term => No. of rows

2. literal in the input part => speed

3. literal in the output part => speed

Terminology and Definition

- literal: a variable or its complement

  Ex: $xy + x'z$ => 4 literals

- cube: product term

  Ex: 4-variable function f (x, y, z, w)

  on-set = {(1, 1, 1, 1)}

  => f = xyzw

  on-set = {(1, 1, 1, 1)
  (0, 1, 1, 1)}

  => f = yzw

  on-set = {(1, 1, 1, 1)
  (1, 0, 1, 1)
  (0, 1, 1, 1)
  (0, 0, 1, 1)}

  => f = zw

Terminology and Definition

- implicant of a function $F(f, d, r)$:

  $$\text{cube } c \subseteq f + d$$

  where f, d, r are on-set, don't-care set, and off-set.
- prime implicant: removing any literal from c will cause $c \cap r \neq \varnothing$
- essential prime implicant: a prime implicant of F which contains a minterm F not contained in any other prime
- irredundant cover: a cover C (union of implicant) that no proper subset of C is also a cover

Minimization of Two-Level Logic

- Algebraic manipulation

  Cout = XY'Cin + X'YCin + XYCin' + XYCin

  = YCin + XCin + XY

  Uniting Theorem

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

  Idempotent Law

  A + A = A

Minimization of Two-Level Logic

- K-map simplification

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

Exact Minimization of Two-Level Logic

- Quine-McClusky

  (1) Generate all primes

  (2) Find a minimum cover

Quine-McClusky

(1) Generate all primes

(utilize $AB+AB'=A(B+B')=A$)

- Grouping cubes according the number of 1's in a cube
- Merging cubes in the neighboring groups and marking the cubes that are merged

Quine-McClusky

Example: utilize AB+AB’=A(B+B’)=A

$$f = \Sigma m\ (\ 4, 5, 6, 8, 9, 10, 13\ ) + d(\ 0, 7, 15\ )$$

0000	0-00	01--
0100	-000	-1-1
1000	010-	
0101	01-0	
0110	100-	
1001	10-0	
1010	01-1	
0111	-101	
1101	011-	
1111	1-01	
	-111	
	11-1	

Quine-McClusky

(2) Select a subset of primes

$$f(x, y, z, w) = x'z'w' + y'z'w' + xy'z' + xy'w' + xz'w + x'y + yw$$

=> the selected sum for f is

$$f(x, y, z, w) = xy'w' + xz'w + x'y$$

A subset of implicant is a cover of the function if each minterm for which the function is 1 is included in at least one implicant of the subset.

Covering Problem

- Define a constraint matrix
  - –column : corresponds to a prime
  - –row : a minterm
  - –A_{ij} = 1 if jth column cover i minterm

	P_1=x'y	P_2=x'z'	P_3=y'z'	P_4=yz
x'y'z'	0	1	1	0
x'yz'	1	1	0	0
x'yz	1	0	0	1
xyz	0	0	0	1
xy'z'	0	0	1	0

Find a subset of columns of minimum cost that covers all rows. (minimum column cover)

P1 + P2 + P3 a cover ?

P1 + P3 + P4 a cover ?

P2 + P3 + P4 a cover ?

Unate Covering

	P_1=x'y	P_2=x'z'	P_3=y'z'	P_4=yz
x'y'z'	0	1	1	0
x'yz'	1	1	0	0
x'yz	1	0	0	1
xyz	0	0	0	1
xy'z'	0	0	1	0

- The constraint matrix can be written as switching function.
  - –interpret "P_i=1" as "column P is selected"
  - –constraint equation

$$(P_2 + P_3)(P_1 + P_2)(P_1 + P_4)P_4P_3=1$$

- A formula where no letter appears with both phases is called unate e.g. xy' + zy' otherwise binate e.g.. xy' + zy
- The formula we obtain is unate. Therefore, the covering problem is called unate covering.

Quine-McClusky

1. Reduction of constraint matrix

Ex:

	$P_1=x'y$	$P_2=x'z'$	$P_3=y'z'$	$P_4=yz$
x'y'z'	0	1	1	0
x'yz'	1	1	0	0
x'yz	1	0	0	1
xyz	0	0	0	1
xy'z'	0	0	1	0

2. Enumeration in the case of cyclic core

cyclic core:

	P_1	P_2	P_3	P_4
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

Three Forms of Reduction

- Elimination of rows covered by essential column
- Elimination of rows through row dominance
- Elimination of columns through column dominance
- Iterate the above three forms of reductions

Reduction

1. Elimination of rows covered by essential column:

   If a row of the constraint matrix is a singleton, the corresponding column must be part of a solution.

   Ex:

	$P_1=x'y$	$P_2=x'z'$	$P_3=y'z'$	$P_4=yz$
x'y'z'	0	1	1	0
x'yz'	1	1	0	0
x'yz	1	0	0	1
~~xyz~~	~~0~~	~~0~~	~~0~~	~~1~~
xy'z'	0	0	1	0

Reduction

	$P_1=x'y$	$P_2=x'z'$	$P_3=y'z'$	$P_4=yz$
$x'y'z'$	0	1	1	0
$x'yz'$	1	1	0	0
$x'yz$	1	0	0	1
xyz	0	0	0	1
$xy'z'$	0	0	1	0

$$(P_2+P_3)(P_1+P_2)(P_1+P_4) P_4 P_3=1$$

if $P_3=1$, $P_4=1$ (must be selected)

$$(P_2 + 1)(P_1 + P_2)(P_1 + 1)11=1$$

$$P_1 + P_2 = 1$$

covers $P_3\ P_4\ P_1$ or $P_3\ P_4\ P_2$ are solutions

Reduction

2. Row or constraint dominance

   If row r_i of the constraint matrix has all ones of another row r_j, r_i is covered whenever r_j is covered. (r_i dominates r_j)

Ex:

	P_1	P_2	P_3	P_4	P_5	P_6
~~1~~	~~1~~	~~1~~				~~1~~
2	1	1				
3		1	1			
~~4~~		~~1~~	~~1~~	~~1~~		
5				1	1	
~~6~~				~~1~~	~~1~~	~~1~~

- Dominating row is deleted

Reduction

Ex:

	P_1	P_2	P_3	P_4	P_5	P_6
~~1~~	~~1~~	~~1~~				~~1~~
2	1	1				
3		1	1			
~~4~~		~~1~~	~~1~~	~~1~~		
5				1	1	
~~6~~				~~1~~	~~1~~	~~1~~

$(P_1+P_2+P_6)(P_1+P_2)(P_2+P_3)(P_2+P_3+P_4)(P_4+P_5)(P_4+P_5+P_6) = 1$

$\Rightarrow \; (P_1+P_2)(P_2+P_3)(P_4+P_5) = 1$

utilize absorption property

$x(x + y) = x$

Reduction

3. Column or variable dominance

The cost of column :

- –each column (prime) corresponds to one AND gate in a SOP form.
- –if the number of gate is the only concern, it is correct to assign the same cost to all columns
- –if the literal is more important

$P_1 = xyz \qquad P_2 = wz$

$cost(P_1) > cost(P_2)$

The total cost is the sum of the cost of the selected column

Reduction

A column P_i has all ones of another column P_j and the cost of P_i is not greater than P_j . We can discard P_j from the matrix

(P_i dominates P_j)

Ex:

	P_1	P_2	P_3	P_4	P_5
2	1	1			
3		1	1		
5				1	1

	P_2	P_4
2	1	
3	1	
5		1

- Dominated column is deleted

Reduction

	P_1	P_2	P_3	P_4	P_5
2	1	1			
3		1	1		
5				1	1

$$(P_1 + P_2)(P_2 + P_3)(P_4 + P_5) = 1$$

P_1 is not selected:

$$F_{P_1=0} = P_2(P_2 + P_3)(P_4 + P_5)$$

P_1 and P_3 are not selected:

$$(F_{P_1=0})_{P_3=0} = P_2 (P_4 + P_5)$$

Enumeration

Enumeration in the case of cyclic core

cyclic core:

	P_1	P_2	P_3	P_4
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

Use divide and conquer strategy

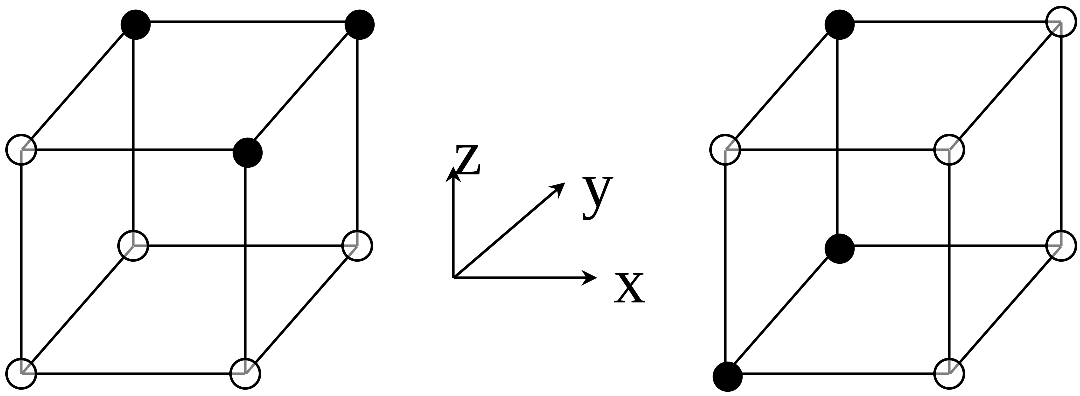
$P_i = 0$ => reduce matrix => find a solution

$P_i = 1$ => reduce matrix => find a solution

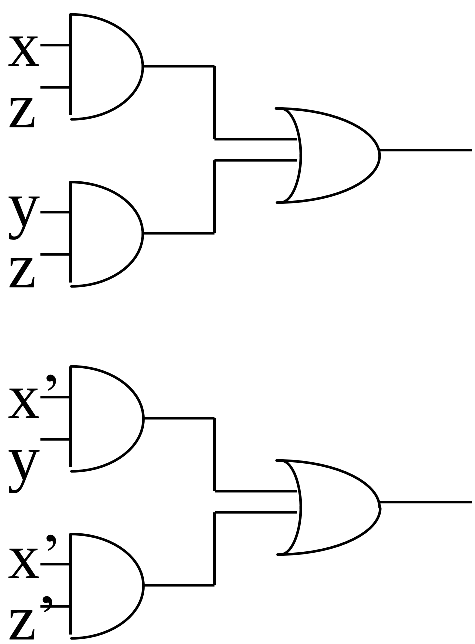
select the smaller one

branch and bound

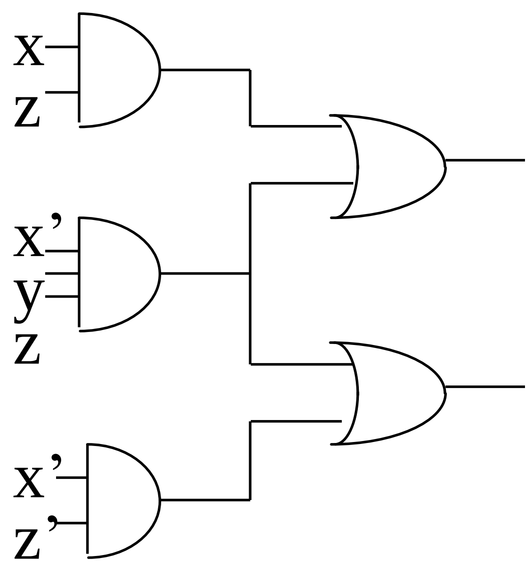
Multiple Output Functions



Two Implementations for the Multiple-Output Function



(a)
no sharing



(b)
sharing

Multiple-output Primes

- Multiple-output primes include the primes of products of the individual functions.

Ex: f1, f2, f3 consider primes of f1, f2, f3,

f1f2, f2f3, f1f3, f1f2f3

Need to consider 2^n - 1 functions

Cubical Representation for Multiple-output

f1 = xz + x'yz

f2 = x'yz + x'z'

x y z	f1 f2
1 - 1	1 0
0 1 1	1 1
0 - 0	0 1

x :1

x' : 0

don't care : -

Tabular Method Applied to the Multiple-output

(1) Two adjacent implicant are merged

=> Their output parts are intersected

(2) Marking the implicant

=> Output part of the new implicant is the same

000	01	0-0	01
010	01	01-	01
011	11	-11	10
101	10	1-1	10
111	10		

Formulating the Covering Problem

- Minterm appears once for each output

$$
\begin{array}{l|l}
P1 = 011 & 11 \\
P2 = 0\text{-}0 & 01 \\
P3 = 01\text{-} & 01 \\
P4 = \text{-}11 & 10 \\
P5 = 1\text{-}1 & 10
\end{array}
$$

$$
\begin{array}{l|l c}
 & & \begin{array}{ccccc} P1 & P2 & P3 & P4 & P5 \end{array} \\
000 & 01 & \\
010 & 01 & \\
011 & 01 & \\
011 & 10 & \\
101 & 10 & \\
111 & 10 &
\end{array}
\begin{bmatrix}
0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
$$