Quine-McCluskey (Tabular) Minimization

Quine-McCluskey (Tabular) Minimization

Two step process utilizing tabular listings to:

- Identify prime implicants (implicant tables)
- Identify minimal PI set (cover tables)

All work is done in tabular form

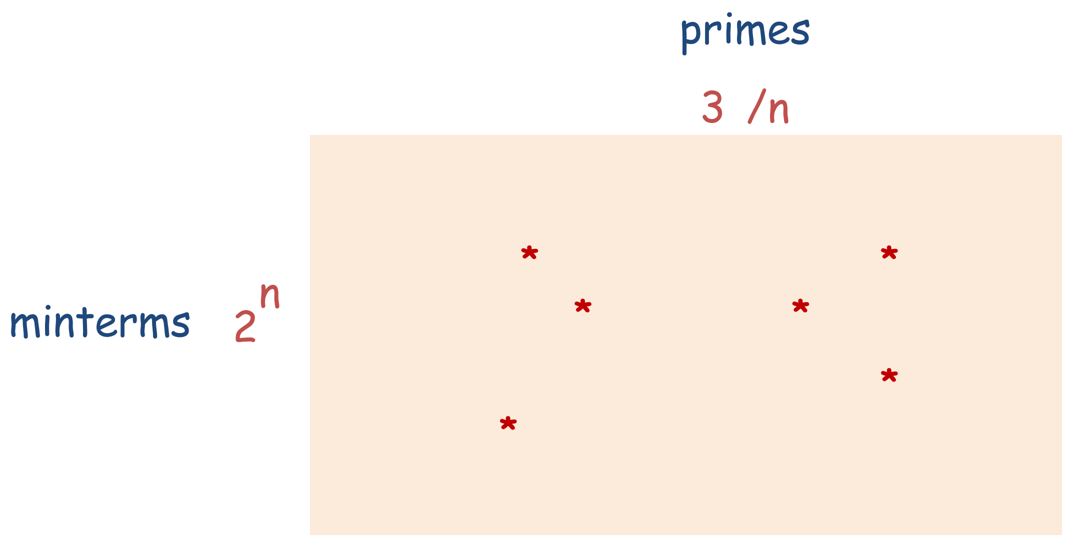
- Number of variables is not a limitation
- Basis for many computer implementations
- Don’t cares are easily handled

Proper organization and term identification are key factors for correct results

Difficulty

Note: Can be

- 2n minterms
- ~ 3n/n primes



✓ Thus $O(2^n)$ rows and $O(3^n/n)$ columns and minimum covering problem is NP-complete.

Example

$F = \bar{w}\bar{x}\bar{y}\bar{z} + \bar{w}x\bar{y}z + \bar{w}xy\bar{z} + wx\bar{y}z$

$d = x\bar{y}\bar{z} + w\bar{y}\bar{z} + xyz + \bar{x}y\bar{z} + wy\bar{z} + \bar{w}\bar{x}\bar{y}z$

Karnaugh map

wx \ yz; groups labelled $\bar{w}\bar{y}$, $\bar{z}$, x

1	d	0	d
d	1	d	1
d	1	d	d
d	0	0	d

	$\bar{z}$	x	$\bar{w}\bar{y}$
$\bar{w}\bar{x}\bar{y}\bar{z}$	1	0	1
$\bar{w}x\bar{y}z$	0	1	1
$\bar{w}xy\bar{z}$	1	1	0
$wx\bar{y}z$	0	1	0

$\sum_c = \bar{z} + x + \bar{w}\bar{y}$

Covering Table

Solution: $\sum_m = \bar{z} + x$

(also $x + \bar{w}\bar{y}$)

Covering Table

Minterms of f (rows); Primes of f+d (columns)

	$\bar{z}$	x	$\bar{w}\bar{y}$
$\bar{w}\bar{x}\bar{y}\bar{z}$	1	0	1
$\bar{w}x\bar{y}z$	0	1	1
$\bar{w}xy\bar{z}$	1	1	0
$\bar{w}\bar{x}\bar{y}\bar{z}$	0	1	0

Row singleton (essential minterm)

Essential prime

- ✓ Definition: An essential prime is any prime that uniquely covers a minterm of f.

Quine-McCluskey Minimization (cont.)

- ✓ Terms are initially listed **one per line in groups**
  - **Each group contains terms with the same number of true and complemented variables**
  - Terms are listed in numerical order within group
- ✓ Terms and implicants are identified using one of three common notations
  - full variable form
  - cellular form
  - 1,0,- form

Example of Different Notations

$F(A, B, C, D) = \sum_m (4,5,6,8,10,13)$

	Full variable	Cellular	1,0,-
1	$\bar{A}B\bar{C}\bar{D}$	4	0100
	$A\bar{B}\bar{C}\bar{D}$	8	1000
2	$\bar{A}B\bar{C}D$	5	0101
	$\bar{A}BC\bar{D}$	6	0110
	$A\bar{B}C\bar{D}$	10	1010
3	$AB\bar{C}D$	13	1101

Notation Forms

- ✓ Full variable form: variables and complements in algebraic form
  - hard to identify when adjacency applies
  - very easy to make mistakes

- ✓ Cellular form: terms are identified by their decimal index value
  - easy to tell when adjacency applies; indexes must differ by **a power of two (one bit)**

- ✓ 1,0,- form: terms are identified by their binary index value
  - **easier to translate to/from full variable form**
  - **easy to identify when adjacency applies, one bit is different**
  - **shows variable(s) dropped when adjacency is used**

- ✓ Different forms may be mixed during the minimization

Implication Table (1,0,-)

✓ Quine-McCluskey Method

- Tabular method to systematically find all prime implicants
  $f(A,B,C,D) = \Sigma_m(1,2.5,6,7,9,10) + \Sigma_d(0,13,15)$
- Part 1: Find all prime implicants
- Step 1: Fill Column 1 with onset and DC-set minterm indices. Group by number of true variables (# of 1's).

NOTE THAT DCs ARE INCLUDED IN THIS STEP!

Implication Table		
Column I		
0000		
0001		
0010		
0101		
0110		
1001		
1010		
0111		
1101		
1111		

Minimization - First Pass (1,0,-)

✓ Quine-McCluskey Method

- Tabular method to systematically find all prime implicants
- $f(A,B,C,D) = \Sigma_m(1,2.5,6,7,9,10) + \Sigma_d(0,13,15)$
- Part 1: Find all prime implicants
- Step 2: Apply Adjacency - Compare elements of group with N 1's against those with N+1 1's. One bit difference implies adjacent. Eliminate variable and place in next column.

  E.g., 0000 vs. 0100 yields 0-00

  0000 vs. 1000 yields -000

  When used in a combination, mark with a check. If cannot be combined, mark with a star. These are the prime implicants.

  Repeat until nothing left.

Implication Table

Column I		Column II		
0000	0	000-	0,1	
		00-0	0,2	
0001	1			
0010	2	0-01	1,5	
		-001	1,9	
0101	5	0-10	2,6	
0110	6	-010	2,10	
1001	9			
1010	10	01-1	5,7	
		-101	5,13	
0111	7	011-	6,7	
1101	13	1-01	9,13	
1111	15	-111	7,15	
		11-1	13,15	

Minimization - Second Pass (1,0,-)

✓ Quine-McCluskey Method

- Step 2 cont.: Apply Adjacency - Compare elements of group with N 1's against those with N+1 1's. One bit difference implies adjacent. Eliminate variable and place in next column.

  E.g., 0000 vs. 0100 yields 0-00

  00-0 vs. 10-0 yields -0-0
- When used in a combination, mark with a check ✓.
- If cannot be combined, mark with a star *. **THESE ARE THE PRIME IMPLICANTS.**

  Repeat until nothing left.

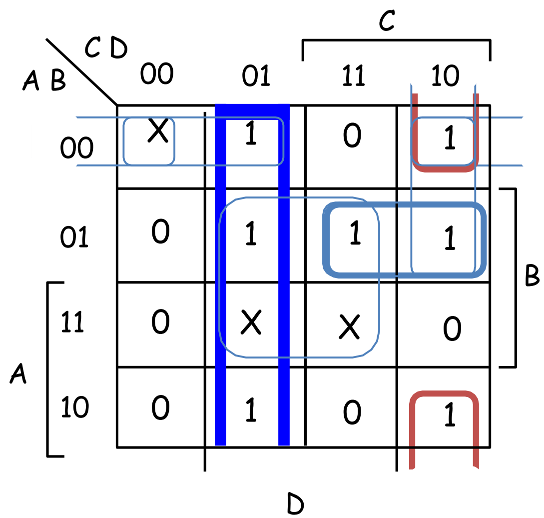
✓ The set of * constitutes the Complete Sum Σ_c

Implication Table

Column I		Column II		Column III	
0000 ✓	0	000-	0,1	--01	1,5,9
		00-0	0,2	- 1-1	5,7,1
0001 ✓	1				
0010 ✓	2	0-01	1,5		
		-001	1,9		
0101 ✓	5	0-10	2,6		
0110 ✓	6	-010	2,10		
1001 ✓	9				
1010 ✓	10	01-1	5,7		
		-101	5,13		
0111 ✓	7	011-	6.7		
1101 ✓	13	1-01	9,13		
1111 ✓	15	-111	7,15		
		11-1	13,15		

Prime Implicants

$f(A,B,C,D) = \Sigma_m (1,2,5,6,7,9,10) + \Sigma_d (0,13,15)$



Prime Implicants:

$000- = \bar{A}\,\bar{B}\,\bar{C}$

$00-0 = \bar{A}\,\bar{B}\,\bar{D}$

$0-10 = \bar{A}\,C\,\bar{D}$

$-010 = \bar{B}\,C\,\bar{D}$

$011- = \bar{A}\,B\,C$

$-1-1 = B\,D$

$--01 = \bar{C}\,D$

Stage 2: find smallest set of prime implicants that cover the active-set

Note that essential prime implicants must be in the final expression

Coverage Table

rows = prime implicants
columns = ON-set elements (minterms)

place an "X" if ON-set element is covered by the prime implicant

NOTE: <u>DON'T</u> INCLUDE DCs IN COVERAGE TABLE; THEY DON'T HAVE TO BE MANDATORY COVERED

Coverage Chart

minterms

primes		1	2	5	6	7	9	10
0,1	000-	X						
0,2	00-0		X					
2,6	0-10		X		X			
2,10	-010		X					(X)
6,7	011-				X	X		
1,5,9,13	--01	X		X			(X)	
5,7,13,15	-1-1			X		X		

Row and Column Dominance

- ✓ Definition: Given two rows i_1 and i_2, a row i_1 is said to dominate i_2 if it has checks in all columns in which i_2 has checks, i.e. it is a superset of i_2

Example:

i_1	x	x	x	x	x	x
i_2	x	x		x	x	

i_1 dominates i_2

- ✓ We can remove row i_2, because we would never choose i_2 in a minimum cover since it can always be replaced by i_1 (i_2 is anymore a prime implicant).

DOMINATED ROWS (**IMPLICANTS**) CAN BE ELIMINATED

Row and Column Dominance

- ✓ Definition: Given two colums j_1 and j_2, if the set of primes of column j_2 is contained in the set of primes of column j_1

Example:

j1	j2
×	
×	×
×	
×	×

j_2 dominates j_1

- ✓ We can remove column j_1 since we have to choose a prime to cover j_2, any such prime also covers j_1, that would result covered as well.

DOMINATED COLUMNS (**MINTERMS**) CAN BE ELIMINATED

Pruning the Covering Table

1. Remove all rows covered by essential primes (columns in row singletons). Put these primes in the cover G.
2. Group identical rows together and remove dominated rows.
3. Remove dominating columns. For equal columns, keep just one to represent them.
4. Newly formed row singletons define n-ary essential primes.
5. Go to 1 if covering table decreased.

- ✓ The algorithm may terminate successfully with a set of primes and an emty table.
- ✓ In case it terminate with a non empty table, the resulting reduced covering table is called the **cyclic core**. This has to be solved. A minimum solution for the cyclic core must be added to the resulting G.

Coverage Table (cont.)

Coverage Chart

		1	2	5	6	7	9	10
0,1	000-	X						
0,2	00-0		X					
2,6	0-10		X		X			
2,10	-010		X					X
6,7	011-				X	X		
1,5,9,13	--01	X		X			X	
5,7,13,15	-1-1			X		X		

		1	2	5	6	7	9	10
0,1	000-	X						
0,2	00-0		X					
2,6	0-10		X		X			
2,10	-010		X					X
6,7	011-				X	X		
1,5,9,13	--01	X		X			X	
5,7,13,15	-1-1			X		X		

If column has a single x, than the implicant associated with the row is essential.
It must appear in the minimum cover

Coverage Table (cont.)

		1	2	5	6	7	9	10
0,1	000-	X						
0,2	00-0		X					
2,6	0-10		X		X			
2,10	**-010**		X					(X)
6,7	011-				X	X		
1,5,9,13	**--01**	X		X			(X)	
5,7,13,15	-1-1			X		X		

Eliminate all columns covered by essential primes

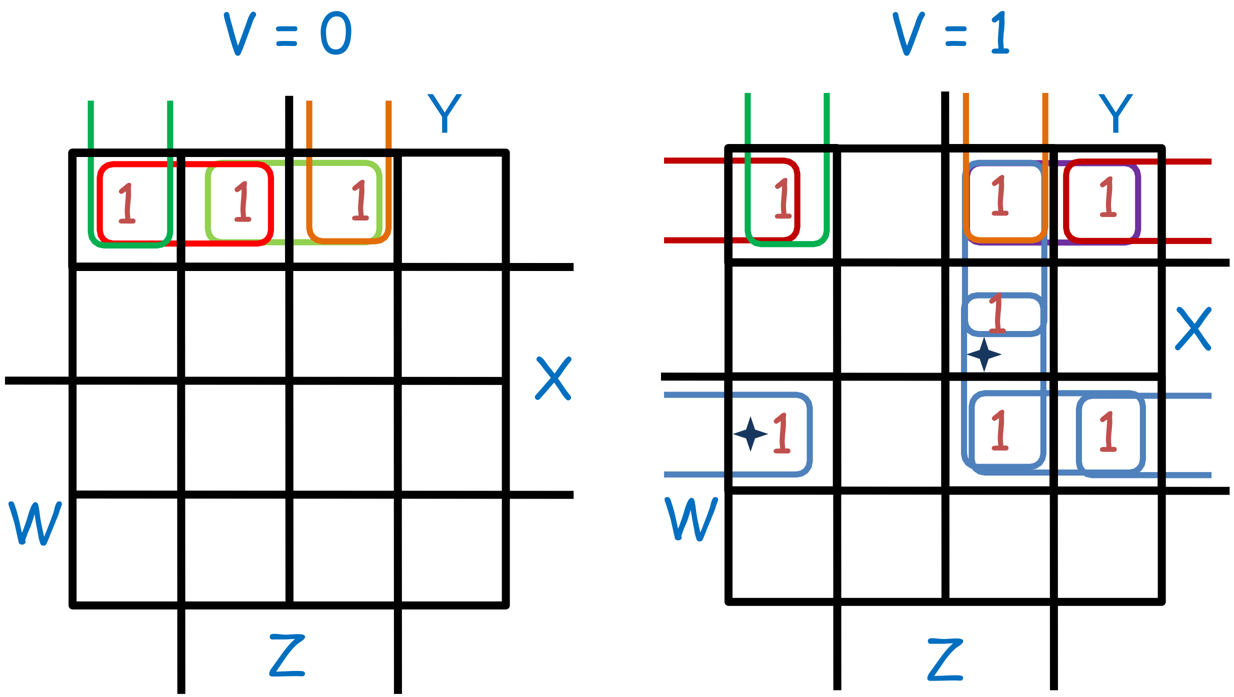
		1	2	5	6	7	9	10
0,1	000-	X						
0,2	00-0		X					
2,6	0-10		X		X			
2,10	**-010**		X					(X)
6,7	**011-**				X	X		
1,5,9,13	**--01**	X		X			(X)	
5,7,13,15	-1-1			X		X		

Find minimum set of rows that cover the remaining columns

$$F = \bar{B}C\bar{D} + \bar{A}BC + \bar{C}D$$

Quine Mc Clunskey: Cyclic Core example

$F=\sum_m (0,1,3,16,18,19,23,28,30,31)$



$F=v'w'y'x'+v'w'x'z+w'x'y'z'+w'x'yz+vw'x'z'+vw'x'y+vwxz'+vwxy+vw'yz+vxyz$

A B C D E F G H I J

Implication Table (1,0,-)

✓ Quine-McCluskey Method

- Tabular method to systematically find all prime implicants
- $f(v,w,x,y,z) = \Sigma m(0,1,3,16,18,19,23,28,30,31)$
- Part 1: Find all prime implicants
- Step 1: Fill Column 1 with active-set and DC-set minterm indices. Group by number of true variables (# of 1's).

Implication Table

	Column I	Column II	
0	00000	0000-	A: 0
1	00001	-0000	C: 0 1
16	1 0000	000-1	B: 1
3	00 011	100-0	E:16 18
18	1001 0	-0011	D: 3 19
19	100 11	1001-	F: 18 19
28	1 1100	10-11	I: 19 23
23	10 111	111-0	G:28 30
30	111 1 0	1-111	J: 23 31
31	1 111 1	1111-	H: 30 31

$F = v'w'y'x' + v'w'x'z + w'x'y'z' + w'x'yz + vw'x'z' + vw'x'y + vwxz' + vwxy + vw'yz + vxyz$

A B C D E F G H I J

Quine Mc Clunskey

$F = \sum_5(0,1,3,16,18,19,23,28,30,31)$

	0	1	3	16	18	19	23	28	30	31
A	X	X								
B		X	X							
C	X			X						
D			X			X				
E				X	X					
F					X	X				
G								X	X	
H									X	X
I						X	X			
J							X			X

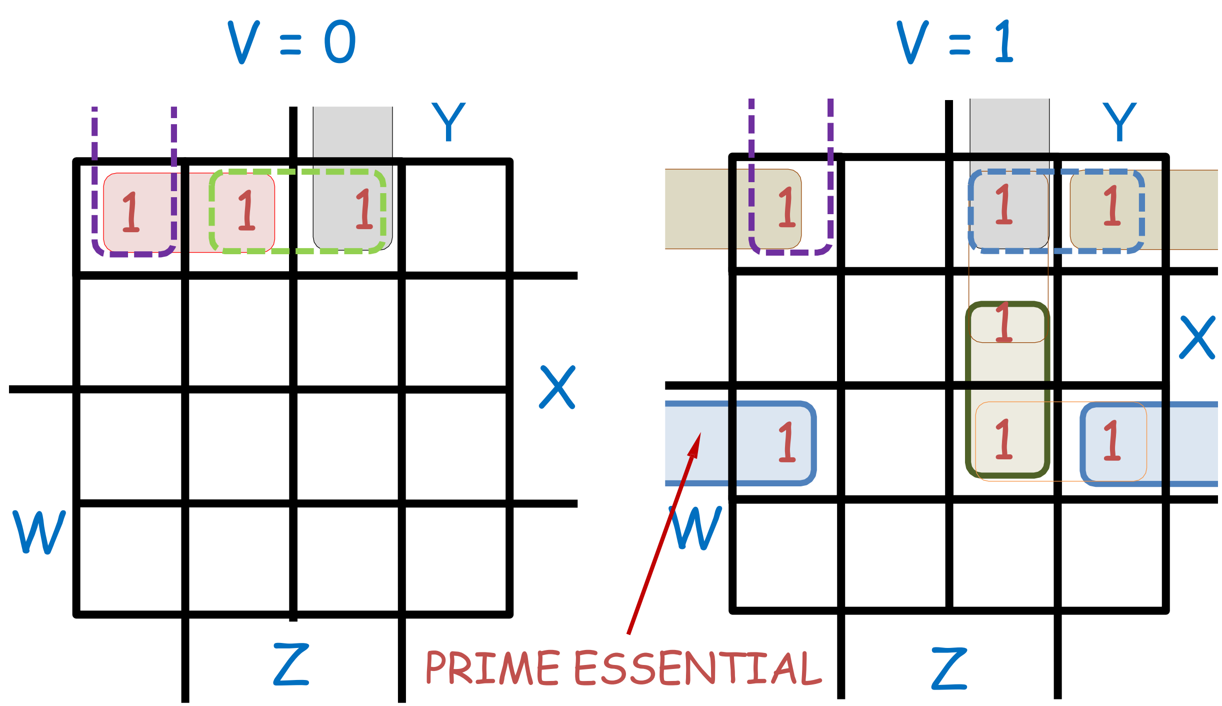
G+J+A+D+E;

Quine Mc Clunskey: Cyclic Core example

$F=\sum_m(0,1,3,16,18,19,23,28,30,31)$

G:28 3
J:23 3

A: 0
D: 3 1
E:16 1



F= v'w'y'x'+v'w'x'z+w'x'y'z'+w'x'yz+vw'x'z'+vw'x'y+**vwxz'**+vwxy+vw'yz+**vxyz**

A B C D E F G H I J

G+J+A+D+E;

Generating Primes – multiple outputs

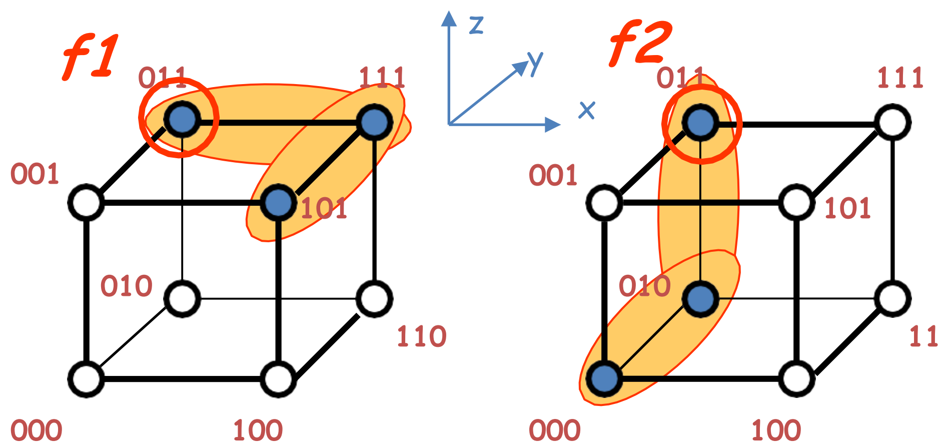
Example: $f1(x, y, z) = \sum m(3,5,7)$, $f2(x, y, z) = \sum m(0,2,3)$

x y z	
0 1 1	- 1 1
1 0 1	1 - 1
1 1 1	

$f1 = yz + zx$

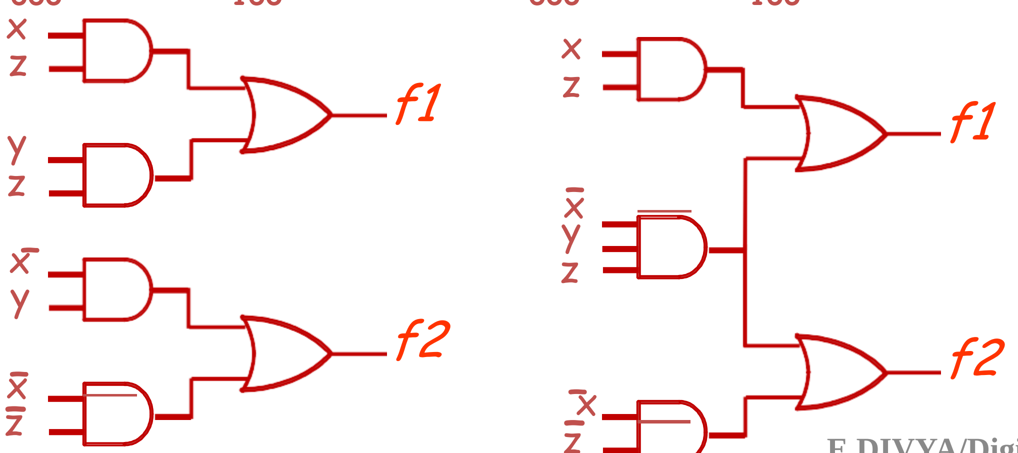
x y z	
0 0 0	0 - 0
0 1 0	0 1 -
0 1 1	

$f2 = \bar{x}\bar{z} + \bar{x}y$



The idea is that we can share terms:

- using separate optimizations 6 gates and G=12
- sharing a term 5 gates and G=11.



$f1\ f2 = (yz+zx)(\bar{x}\bar{z}+\bar{x}y)$

$f1\ f2 = \bar{x}yz$

Generating Primes – multiple outputs

- Theorem: if p_1 is a prime implicant for f_1, and p_2 is a prime implicant for f_2, then if $p_1.p_2 \neq 0$, $p_1.p_2$ is a prime implicant of $f_1.f_2$
- Theorem: if p_3 is a prime implicant for $f_1.f_2$, then there exist p_1 for f_1, and p_2 for f_2, such that $p_3 = p_1.p_2$
- We can conclude that all prime implicants of $f_1.f_2$ are minimal sharable products for f_1 and f_2: and that all prime implicants for $f_1.f_2$ are created by products of prime implicants for f_1 and f_2
- The way to use this is to make the prime implicants of $f_1.f_2$ available to the minimizations of f_1 and f_2 by extending the table concept

Generating Primes – multiple outputs

✓ Procedure similar to single-output function, except: **include also the primes of the products of individual functions**

	f_1 minterms	f_2 minterms
Rows for f_1 prime implicants: mark only f_1 columns		
Rows for f_2 prime implicants: mark only f_2 columns		
Rows for $f_1 f_2$ prime implicants: mark both f_1 and f_2 columns		

Minimize multiple-output cover

✓ Example, cont.

f_1

$m_3 = 0\ 1\ 1$ $p_1 = y\,z$

$m_5 = 1\ 0\ 1$ $p_2 = x\,z$

$m_7 = 1\ 1\ 1$

f_2

$m_0 = 0\ 0\ 0$ $p_3 = \bar{x}\,y$

$m_2 = 0\ 1\ 0$ $p_4 = \bar{x}\,\bar{z}$

$m_3 = 0\ 1\ 1$

f_2 $m_3 = 0\ 1\ 1$ $p_5 = \bar{x}\,y\,z$

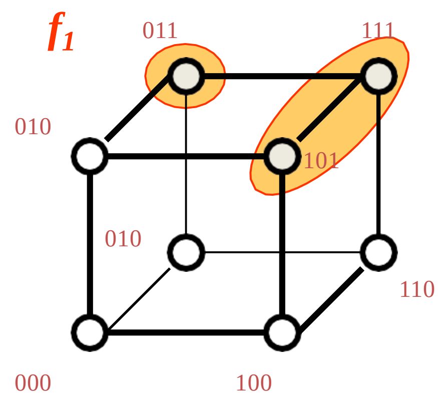
	m_3	m_5	m_7	m_0	m_2	m_3
p_1	✓		✓			
p_2		✓	✓			
p_3					✓	✓
p_4				✓	✓	
p_5	✓					✓

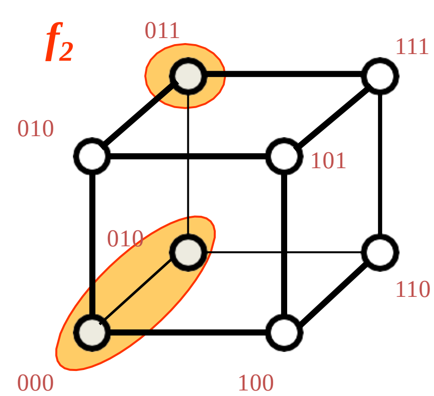
	m_3	m_3
p_1	✓	
p_3		✓
p_5	✓	✓

Note that row p_5 dominates rows p_1 and p_3, removing these rows, the coverage is complete

Min cover has 3 primes:

$F = \{ p_2, p_4, p_5 \}$

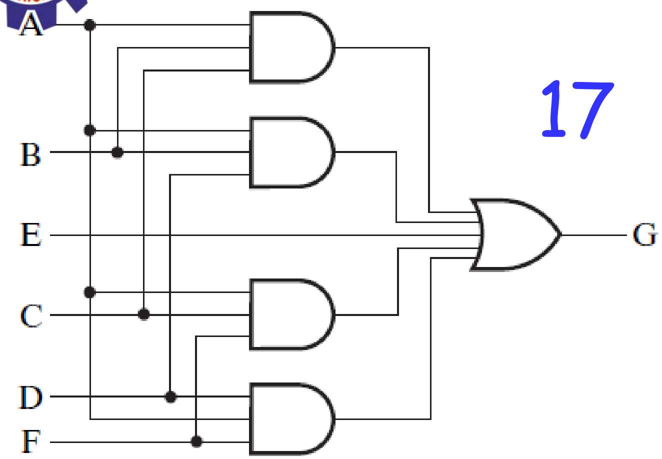




Multiple-Level Optimization

- ✓ Multiple-level circuits: circuits that are not two-levels (with or without input and/or output inverters)
- ✓ **Multiple-level circuits can have reduced gate input cost** compared to two-level (SOP and POS) circuits, obviously **augmenting the execution time**
- ✓ Multiple-level optimization is performed by applying transformations to circuits represented by equations while evaluating cost and execution time

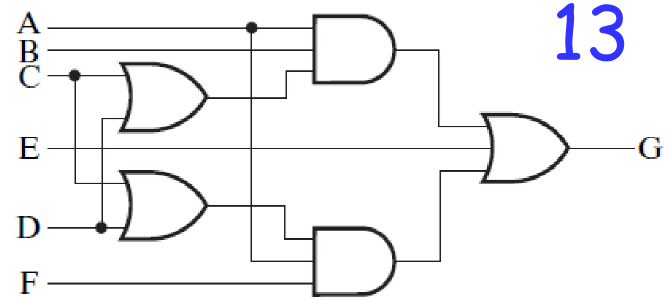
Multi level circuits – circuit analysis



(a)

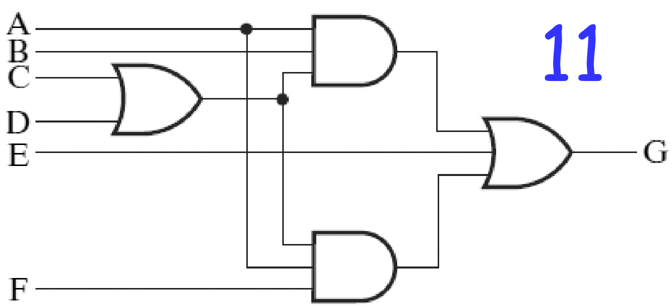
17

$G = ABC+ABD+E+ACF+ADF$



13

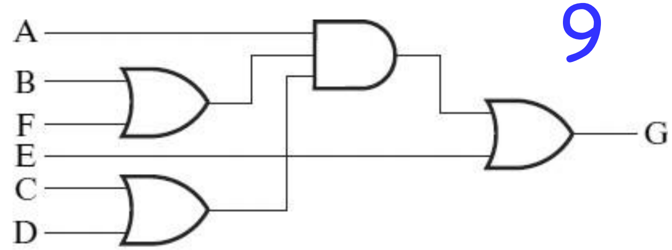
$G = AB(C+D)+E+AF(C+D)$



(c)

11

$G = (C+D)(AB+AF)+E$



(d)

9

$G = A(C+D)(B+F)+E$

Basic Transformations

- ✓ Factoring - finding a factored form from SOP or POS expression
- ✓ Elimination of G into F - expression function F as a function of G and some or all of its original variables
- ✓ Extraction - decomposition applied to multiple functions simultaneously

Transformation Examples

✓ Algebraic Factoring

$F = \overline{A}\,\overline{C}\,\overline{D} + \overline{A}B\overline{C} + ABC + AC\overline{D}$ $G = 16$

- Factoring:

$F = \overline{A}(\overline{C}\,\overline{D} + B\overline{C}) + A(BC + C\overline{D})$ $G = 18$

- Factoring again:

$F = \overline{A}\,\overline{C}(B + \overline{D}) + AC(B + \overline{D})$ $G = 12$

- Factoring again:

$F = (\overline{A}\,\overline{C} + AC)(B + \overline{D})$ $G = 10$

Transformation Examples

Elimination

Beginning with two functions: $X = B + C$ $Y = A + B$

$Z = \overline{A}X + CY$ $G = 10$

- Eliminating X and Y from Z:

$Z = \overline{A}(B + C) + C(A + B)$ $G = 10$

- "Flattening" (Converting to SOP expression):

$Z = \overline{A}B + \overline{A}C + AC + BC$ $G = 12$

- This has increased the cost, but has provided a new SOP expression for two-level optimization.
- Two-level Optimization

$Z = \overline{A}B + C$ $G = 4$

- Increasing gate input count G temporarily can result in a final solution with a smaller G

Transformation Examples

✓ Extraction

- Beginning with two functions:

$E = \overline{A}\,\overline{B}\,\overline{D} + \overline{A}BD$

$H = \overline{B}C\overline{D} + BCD$ $\quad G = 16$

- Finding a common factor and defining it as a function:

$F = \overline{B}\,\overline{D} + BD$

- We perform extraction by expressing E and H as the three functions:

$E = \overline{A}F,\ H = CF$ $\quad G = 10$

- The reduced cost G results from the sharing of logic between the two output functions

THANK YOU