Minimization of Switching Functions

Zvi Kohavi and Niraj K. Jha

Simplifying Switching Functions

Finding an equivalent switching expression that minimizes some cost criteria:

1. Minimize literal count
2. Minimize literal count in sum-of-products (or product-of-sums) expression
3. Minimize number of terms in a sum-of-products expression provided no other expression exists with the same number of terms and fewer literals

Example: $f(x,y,z) = x'yz' + x'y'z + xy'z' + x'yz + xyz + xy'z$
$= x'z' + y'z' + yz + xz$

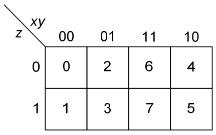
Irredundant sum-of-products expression: no term or literal can be deleted without changing its logical value

- Not necessarily minimal: $f(x,y,z) = x'z' + xy' + yz$
- Not necessarily unique: $f(x,y,z) = x'y + y'z' + xz$

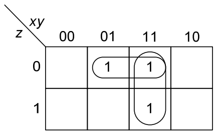
The Map Method

Algebraic procedure to combine terms using the *Aa* + *Aa'* = *A* rule

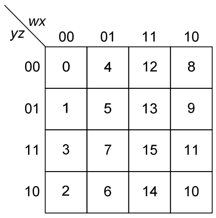
Karnaugh map: modified form of truth table



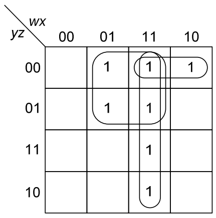
(*a*) Location of minterms in a three-variable map.



(*b*) Map for function *f*(*x*,*y*,*z*) =∑(2,6,7) = *yz′* + *xy*.



(*c*) Location of minterms in a four-variable map.



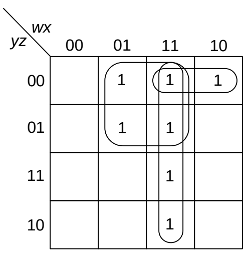
(*d*) Map for function *f*(*w*,*x*,*y*,*z*) =∑(4,5,8,12,13,14,15) = *wx* + *xy′* + *wy′z′*.

Simplification and Minimization of Functions

Cube: collection of 2^m cells, each adjacent to *m* cells of the collection

- Cube is said to cover these cells
- Cube expressed by a product of *n-m* literals for a function containing *n* variables
- *m* literals not in the product said to be eliminated

Example: *w'xy'z'* + *w'xy'z* + *wxy'z'* + *wxy'z* = *xy'*(*w'z'* + *w'z* + *wz'* + *wz*) = *xy'*



(*d*) Map for function *f*(*w*,*x*,*y*,*z*) =∑(4,5,8,12,13,14,15) = *wx* + *xy′* + *wy′z′*.

Minimization (Contd.)

Example: $f = yz' + xy$

- Use of cell 6 in forming both cubes justified by idempotent law

$z \backslash xy$	00	01	11	10
0	0	2	6	4
1	1	3	7	5

(*a*) Location of minterms in a three-variable map.

$z \backslash xy$	00	01	11	10
0		1	1	
1			1	

(*b*) Map for function $f(x,y,z) = \sum(2,6,7) = yz' + xy$.

Corresponding algebraic manipulations:

$$
\begin{aligned}
f &= x'yz' + xyz' + xyz \\
&= x'yz' + xyz' + xyz' + xyz \\
&= yz'(x' + x) + xy(z' + z) \\
&= yz' + xy
\end{aligned}
$$

5

Minimization (Contd.)

Minimal expression: cover all the 1 cells with the smallest number of cubes such that each cube is as large as possible

- A cube contained in a larger cube must never be selected
- If there is more than one way of covering the map with a minimal number of cubes, select the cover with larger cubes
- A cube contained in any combination of other cubes already selected in the cover is redundant by virtue of the consensus theorem

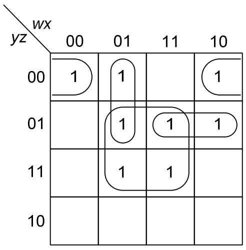
Rules for minimization:

1. First, cover those 1 cells by cubes that cannot be combined with other 1 cells; continue to 1 cells that have a single adjacent 1 cell (thus can form cubes of only two cells)
2. Next, combine 1 cells that yield cubes of four cells, but are not part of any cube of eight cells, and so on
3. Minimal expression: collection of cubes that are as large and as few in number as possible, such that each 1 cell is covered by at least one cube

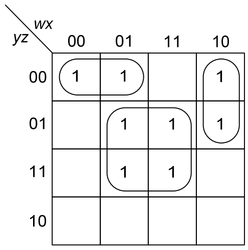
6

Minimization (Contd.)

Example: Two irredundant expressions for $f(w,x,y,z) = \sum(0,4,5,7,8,9,13,15)$



(*a*) $f = x'y'z' + w'xy' + wy'z + xz$
is an irredundant expression.

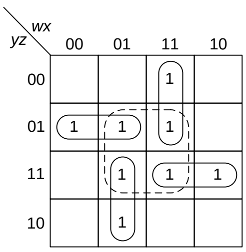


(*b*) $f = w'y'z' + wx'y' + xz$ is the
unique minimal expression.

Minimization (Contd.)

Example: $f(w,x,y,z) = \sum(1,5,6,7,11,12,13,15)$

- Only one irredundant form: $f = wxy' + wyz + w'xy + w'y'z$
- Dotted cube xz is redundant



Minimal Product-of-sums

Dual procedure: product of a minimum number of sum factors, provided there is no other such product with the same number of factors and fewer literals

- Variable corresponding to a 1 (0) is complemented (uncomplemented)
- Cubes are formed of 0 cells

Example: either one of minimal sum-of-products or minimal product-of-sums can be better than the other in literal count

yz \ wx	00	01	11	10
00				
01		1		1
11				
10		1		1

(*a*) Map of $f(x,y,z) = \sum(5,6,9,10)$
$= w'xy'z + wx'y'z + w'xyz' + wx'yz'$.

yz \ wx	00	01	11	10
00	0	0	0	0
01	0	1	0	1
11	0	0	0	0
10	0	1	0	1

(*b*) Map of $f(x,y,z)$
$= \prod(0,1,2,3,4,7,8,11,12,13,14,15)$
$= (y + z)(y' + z')(w + x)(w' + x')$.

Don't-care Combinations

Don't-care combination ϕ: combination for which the value of the function is not specified. Either

- input combinations may be invalid
- precise output value is of no importance

Since each don't-care can be specified as either 0 or 1, a function with k don't-cares corresponds to a class of 2^k distinct functions. Our aim is to choose the function with the minimal representation

- Assign 1 to some don't-cares and 0 to others in order to increase the size of the selected cubes whenever possible
- No cube containing only don't-care cells may be formed, since it is not required that the function equal 1 for these combinations

Code Converter

Example: code converter from BCD to excess-3 code

- Combinations 10 through 15 are don't-cares

Truth table

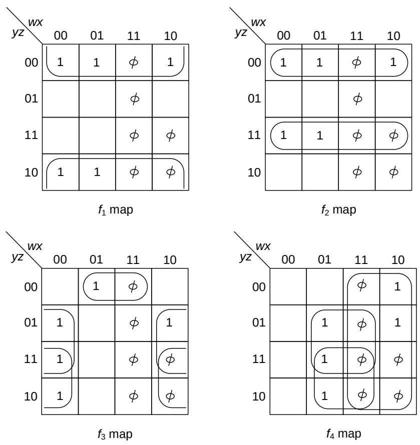
Decimal number	*BCD inputs* w	x	y	z	*Excess-3 outputs* f_4	f_3	f_2	f_1
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

Output functions

$$f_1 = \sum(0,2,4,6,8) + \sum_\phi(10,11,12,13,14,15)$$
$$f_2 = \sum(0,3,4,7,8) + \sum_\phi(10,11,12,13,14,15)$$
$$f_3 = \sum(1,2,3,4,9) + \sum_\phi(10,11,12,13,14,15)$$
$$f_4 = \sum(5,6,7,8,9) + \sum_\phi(10,11,12,13,14,15)$$

11

Code Converter (Contd.)



$f_1 = z'$

$f_2 = y'z' + yz$

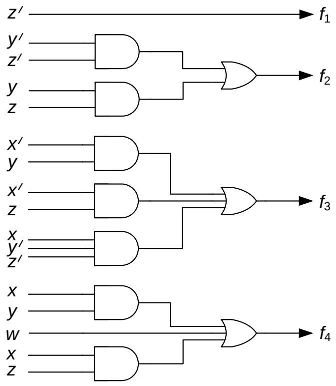
$f_3 = x'y + x'z + xy'z'$

$f_4 = w + xy + xz$

12

Logic Network for Code Converter

Two-level AND-OR realization:



Five-variable Map

General five-variable map:

yz \ vwx	000	001	011	010	110	111	101	100
00	0	4	12	8	24	28	20	16
01	1	5	13	9	25	29	21	17
11	3	7	15	11	27	31	23	19
10	2	6	14	10	26	30	22	18

Example: Minimize $f(v,w,x,y,z) =$
$\sum(1,2,6,7,9,13,14,15,17,22,23,25,29,30,31)$

yz \ vwx	000	001	011	010	110	111	101	100
00								
01	1		1	1	1	1		1
11		1	1			1	1	
10	1	1	1			1	1	

$f(v,w,x,y,z) = x'y'z + wxz + xy + v'w'yz'$

Minimal Functions and Their Properties

Implicants: function *f* covers function *g* with the same input variables if *f* has a 1 in every row of the truth table in which *g* has a 1

- If *f* covers *g* and *g* covers *f*, then *f* and *g* are equivalent
- Let *h* be a product of literals. If *f* covers *h*, then *h* is said to imply *f* or *h* is said to be an implicant of *f*, denoted as *h* -> *f*

Example: If $f = wx + yz$ and $h = wxy'$, then *f* covers *h* and *h* implies *f*

Prime implicant *p* of function *f*: product term covered by *f* such that the deletion of any literal from *p* results in a new product not covered by *f*

- *p* is a prime implicant if and only if *p* implies *f*, but does not imply any product with fewer literals which in turn also implies *f*

Example: $x'y$ is a prime implicant of $f = x'y + xz + y'z'$ since it is covered by *f* and neither x' nor *y* alone implies *f*

Theorem: Every irredundant sum-of-products equivalent to *f* is a union of prime implicants of *f*

Deriving Prime Implicants and Minimal Expressions

Example: for $f(w,x,y,z) = \sum(0,4,5,7,8,9,13,15)$ below, set of prime implicants $P = \{xz, w'y'z', wx'y', x'y'z', w'xy', wy'z\}$

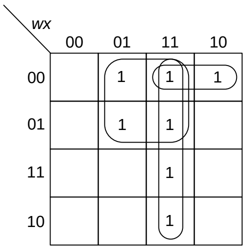
yz \ wx	00	01	11	10
00	1	1		1
01		1	1	1
11		1	1	
10				

Essential prime implicants: covers at least one minterm not covered by any other prime implicant, e.g., *xz*

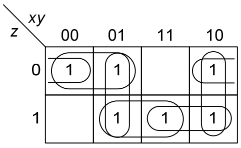
- Since all minterms must be covered, all essential prime implicants must be contained in any irredundant expression of the function

Minimal Expressions (Contd.)

Example: prime implicants of $f(w,x,y,z) = \sum(4,5,8,12,13,14,15)$ are all essential



Example: Cyclic prime implicant chart in which no prime implicant is essential, all prime implicants have the same size, and every 1 cell is covered by exactly two prime implicants

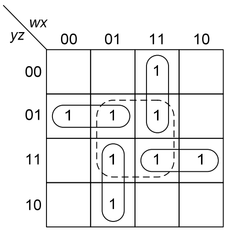


Procedure for Deriving Minimal Sum-of-products Expression

Procedure:

1. Obtain all essential prime implicants and include them in the minimal expression
2. Remove all prime implicants which are covered by the sum of some essential prime implicants
3. If the set of prime implicants derived so far covers all the minterms, it yields a unique minimal expression. Otherwise, select additional prime implicants so that the function is covered completely and the total number and size of the added prime implicants are minimal

Example: prime implicant xz is covered by the sum of four essential prime implicants, and hence xz must not be included in any irredundant expression of the function



Tabulation Procedure for Obtaining the Set of All Prime Implicants

Systematic Quine-McCluskey tabulation procedure: for functions with a large number of variables

- Fundamental idea: repeated application of the combining theorem $Aa + Aa' = A$ on all adjacent pairs of terms yields the set of all prime implicants

Example: minimize $f_1(w,x,y,z) = \sum(0,1,8,9) = w'x'y'z' + w'x'y'z + wx'y'z' + wx'y'z$

- Combine first two and last two terms to yield
  $f_1(w,x,y,z) = w'x'y'(z' + z) + wx'y'(z' + z)$
  $= w'x'y' + wx'y'$
- Combine this expression in turn to yield
  $f_1(w,x,y,z) = x'y'(w' + w)$
  $= x'y'$
- Similar result can be obtained by initially combining the first and third and the second and fourth terms

19

Tabulation Procedure (Contd.)

Two k-variable terms can be combined into a single (k-1)-variable term if and only if they have k-1 identical literals in common and differ in only one literal

- Using the binary representation of minterms: two minterms can be combined if their binary representations differ in only one position

Example: $w'x'y'z$ (0001) and $wx'y'z$ (1001) can be combined into -001, indicating w has been absorbed and the combined term is $x'y'z$

20

Tabulation Procedure (Contd.)

Procedure:

1. Arrange all minterms in groups, with all terms in the same group having the same number of 1's. Start with the least number of 1's (called the index) and continue with groups of increasing numbers of 1's.
2. Compare every term of the lowest-index group with each term in the successive group. Whenever possible, combine them using the combining theorem. Repeat by comparing each term in a group of index i with every term in the group of index $i + 1$. Place a check mark next to every term which has been combined with at least one term.
3. Compare the terms generated in step 2 in the same fashion: generate a new term by combining two terms that differ by only a single 1 and whose dashes are in the same position. Continue until no further combinations are possible. The remaining unchecked terms constitute the set of prime implicants.

Example

Example: apply procedure to $f_2(w,x,y,z) = \sum(0,1,2,5,7,8,9,10,13,15)$

Step (*i*)

	w	*x*	*y*	*z*	
0	0	0	0	0	✓
1	0	0	0	1	✓
2	0	0	1	0	✓
8	1	0	0	0	✓
5	0	1	0	1	✓
9	1	0	0	1	✓
10	1	0	1	0	✓
7	0	1	1	1	✓
13	1	1	0	1	✓
15	1	1	1	1	✓

Step (*ii*)

	w	*x*	*y*	*z*	
0,1	0	0	0	--	✓
0,2	0	0	--	0	✓
0,8	--	0	0	0	✓
1,5	0	--	0	1	✓
1,9	--	0	0	1	✓
2,10	--	0	1	0	✓
8,9	1	0	0	--	✓
8,10	1	0	--	0	✓
5,7	0	1	--	1	✓
5,13	--	1	0	1	✓
9,13	1	--	0	1	✓
7,15	--	1	1	1	✓
13,15	1	1	--	1	✓

Step (*iii*)

	w	*x*	*y*	*z*	
0,1,8,9	--	0	0	--	*A*
0,2,8,10	--	0	--	0	*B*
1,5,9,13	--	--	0	1	*C*
5,7,13,15	--	1	--	1	*D*

$P = \{x'y', x'z', y'z, xz\}$

Tabulation Procedure using Decimal Notation in the Presence of Don't-cares

Example: apply procedure to $f_3(v,w,x,y,z) = \sum(13,15,17,18,19,20,21,23,25,27,29,31) + \sum_{\phi}(1,2,12,24)$

1	✓
2	✓
12	✓
17	✓
18	✓
20	✓
24	✓
13	✓
19	✓
21	✓
25	✓
15	✓
23	✓
27	✓
29	✓
31	✓

(*a*)

1,17	(16)	*H*
2,18	(16)	*G*
12,13	(1)	*F*
17,19	(2)	✓
17,21	(4)	✓
17,25	(8)	✓
18,19	(1)	*E*
20,21	(1)	*D*
24,25	(1)	*C*
13,15	(2)	✓
13,29	(16)	✓
19,23	(4)	✓
19,27	(8)	✓
21,23	(2)	✓
21,29	(8)	✓
25,27	(2)	✓
25,29	(4)	✓
15,31	(16)	✓
23,31	(8)	✓
27,31	(4)	✓
29,31	(2)	✓

(*b*)

17,19,21,23	(2,4)	✓
17,19,25,27	(2,8)	✓
17,21,25,29	(4,8)	✓
13,15,29,31	(2,16)	*B*
19,23,27,31	(4,8)	✓
21,23,29,31	(2,8)	✓
25,27,29,31	(2,4)	✓

(*c*)

17,19,21,23,25,27,29,31 (2,4,8) *A*

(*d*)

P = {*vz, wxz, vwx'y', vw'xy', vw'x'y, v'wxy', w'x'yz', w'x'y'z'*}

Prime Implicant Chart

Prime implicant chart: pictorially displays covering relationships between prime implicants and minterms

Example: prime implicant chart for $f_2(w,x,y,z) = \sum(0,1,2,5,7,8,9,10,13,15)$

	✓ 0	1	✓ 2	✓ 5	✓ 7	✓ 8	9	✓ 10	✓ 13	✓ 15
$A = x'y'$	×	×				×	×			
✓ $B = x'z'$	×		⊗			×		⊗		
$C = y'z$		×		×			×		×	
✓ $D = xz$				×	⊗				×	⊗

Cover: a row is said to cover the columns in which it has x's

Problem: select a minimal subset of prime implicants such that each column contains at least one x in the rows corresponding to the selected subset and the total number of literals in the prime implicants selected is as small as possible

Essential rows: if a column contains a single x, the prime implicant corresponding to the row in which the x appears is essential, e.g., *B, D*

Cover remaining minterms 1 and 9 using *A* or *C:* thus, two minimal expressions: $f_2 = x'z' + xz + x'y'$ or $f_2 = x'z' + xz + y'z$

Don’t-care Combinations

Don’t-cares: not listed as column headings in the prime implicant chart

Example: $f_3(v,w,x,y,z) = \sum(13,15,17,18,19,20,21,23,25,27,29,31) + \sum_{\phi}(1,2,12,24)$

	✓13	✓15	✓17	18	✓19	✓20	✓21	✓23	✓25	✓27	✓29	✓31
✓ $A = vz$			×		×		×	⊗	×	⊗	×	×
✓ $B = wxz$	×	⊗									×	×
$C = vwx'y'$									×			
✓ $D = vw'xy'$						⊗	×					
$E = vw'x'y$				×	×							
$F = v'wxy'$	×											
$G = w'x'yz'$				×								
$H = w'x'y'z$			×									

Selection of nonessential prime implicants facilitated by listing prime implicants in decreasing order of the number of minterms they cover

Essential prime implicants: *A*, *B*, and *D*. They cover all minterms except 18, which can be covered by *E* or *G*, giving rise to two minimal expressions

Determining the Set of All Irredundant Expressions

Deriving the minimal sum-of-products through prime implicant chart inspection: difficult for more complex cases

Example: $f_4(v,w,x,y,z) = \sum(0,1,3,4,7,13,15,19,20,22,23,29,31)$

	0	1	✓3	4	✓7	✓13	✓15	✓19	20	22	✓23	✓29	✓31
✓ $A = wxz$						⊗	×					⊗	×
$B = xyz$					×		×				×		×
✓ $C = w'yz$			×		×			⊗			×		
$D = vw'xy$										×	×		
$E = vw'xz'$									×	×			
$F = w'xy'z'$				×					×				
$G = v'w'x'z$		×	×										
$H = v'w'y'z'$	×			×									
$I = v'w'x'y'$	×	×											

(*a*) Prime implicant chart.

	0	1	4	20	22
D					×
E				×	×
F			×	×	
G		×			
H	×		×		
I	×	×			

(*b*) Reduced prime implicant chart.

While every irredundant expression must contain *A* and *C*, none of them may contain *B* since it covers minterms already covered by *A* and *C*. The reduced chart, obtained after removing *A*, *B*, and *C*, has two x’s in each column

Example (Contd.)

Use propositional calculus: define prime implicant function p to be 1 if each column is covered by at least one of the chosen prime implicants, and 0 if not

$$p = (H + I)(G + I)(F + H)(E + F)(D + E)$$
$$= EHI + EFI + DFI + EGH + DFGH$$

At least three rows are needed to cover the reduced chart: E, H, and I, or E, F, and I, and so on

Since all prime implicants in the reduced chart have the same literal count, there are four minimal sum-of-products:

$f_4(v,w,x,y,z) = A + B + E + H + I = wxz + w'yz + vw'xz' + v'w'y'z' + v'w'x'y'$

$f_4(v,w,x,y,z) = A + B + E + F + I = wxz + w'yz + vw'xz' + w'xy'z' + v'w'x'y'$

$f_4(v,w,x,y,z) = A + B + D + F + I = wxz + w'yz + vw'xy + w'xy'z' + v'w'x'y'$

$f_4(v,w,x,y,z) = A + B + E + G + H = wxz + w'yz + vw'xz' + v'w'x'z + v'w'y'z'$

Reduction of the Chart

Aim: find just one minimal expression rather than all such expressions

Example: $f_5(v,w,x,y,z) =$
$\sum(1,3,4,5,6,7,10,11,12,13,14,15,18,19,20,21,22,23,25,26,27)$

	✓1	✓3	✓4	✓5	✓6	✓7	10	11	✓12	✓13	✓14	✓15	18	19	✓20	✓21	✓22	✓23	✓25	26	✓27
✓ $A = w'x$			×	×	×	×									⊗	⊗	×	×			
✓ $B = v'x$			×	×	×	×			⊗	⊗	×	×									
$C = vx'y$													×	×						×	×
$D = vw'y$													×	×			×	×			
$E = wx'y$							×	×												×	×
$F = v'wy$							×	×			×	×									
$G = x'yz$		×						×						×							×
$H = w'yz$		×				×								×				×			
$I = v'yz$		×				×		×				×									
✓ $J = v'w'z$	⊗	×		×		×															
✓ $K = vwx'z$																			⊗		×

(*a*) Prime implicant chart.

Example (Contd.)

	10	11	18	19	26
C			×	×	×
D			×	×	
E	×	×			×
F	×	×			
G		×		×	
H				×	
I		×			

(*b*) Reduced prime implicant chart.

	✓ 10	✓ 11	✓ 18	✓ 19	✓ 26
✓*C*			×	×	×
✓*E*	×	×			×
G		×		×	

(*c*) Final chart.

Dominated row: row *U* of the chart **dominates** row *V* if *U* covers every column covered by *V*. If *U* does not have more literals than *V* then *V* can be deleted from the chart.

Example: *I*, *D*, and *F* can be deleted because they are dominated by *G*, *C*, and *E*, respectively

From the final chart: $f_5(v,w,x,y,z) = A + B + J + K + C + E$

Dominating Column

	10	11	18	19	26
C			×	×	×
D			×	×	
E	×	×			×
F	×	×			
G		×		×	
H				×	
I		×			

(*b*) Reduced prime implicant chart.

Dominating column: column *i* of the chart **dominates** column *j* if *i* has an x in every row in which *j* has an x. Hence, dominating column *i* can be deleted.

Example: column 11 dominates column 10. In order to cover column 10, either *E* or *F* must be selected, whereby column 11 will also automatically be covered. Similarly, since column 19 covers column 18, column 19 can be deleted.

Final solution is still: $f_5(v,w,x,y,z) = A + B + J + K + C + E$

Branching Method

When chart has no essential prime implicant, dominated row or dominating column: use branching method

Example: $f_6(w,x,y,z) = \sum(0,1,5,7,8,10,14,15)$

yz \ wx	00	01	11	10
00	1			1
01	1	1		
11		1	1	
10			1	1

(*a*) Cyclic map.

	0	1	5	7	8	10	14	15
$A = w'x'y'$	×	×						
$B = w'y'z$		×	×					
$C = w'xz$			×	×				
$D = xyz$				×				×
$E = wxy$							×	×
$F = wyz'$						×	×	
$G = wx'z'$					×	×		
$H = x'y'z'$	×				×			

(*b*) Cyclic prime implicant chart.

	5	7	8	10	14	15
B	×					
C	×	×				
D		×				×
E					×	×
F				×	×	
G			×	×		
H			×			

(*c*) Reduced chart after selection of row *A*.

	1	5	7	10	14	15
A	×					
B	×	×				
C		×	×			
D			×			×
E					×	×
F				×	×	
G				×		

(*d*) Reduced chart after selection of row *H*.

To cover column 0: either *A* or *H* has to be selected

If *A* is arbitrarily chosen: *C* (*G*) dominates *B* (*H*): $f_6(w,x,y,z) = A+C+G+E$

If *H* is arbitrarily chosen: *B* (*F*) dominates *A* (*G*): $f_6(w,x,y,z) = H+B+D+F$

Map-entered Variables

Map-entered variables: entering variables into the map cells in addition to 0, 1 or don't-care. An *n*-variable map can now specify functions of more than *n* variables

z \ xy	00	01	11	10
0	0	*A*	*A*	*C*
1	*B*	1	ϕ	B'

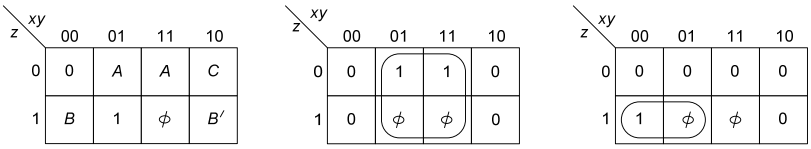
(*a*) Initial map.

Product corresponding to cell 010: $Ax'yz'$

Procedure for Covering Map-entered Variables

Procedure:

1. Treat all map-entered variables as 0's and find a minimal expression for the resulting map
2. To cover a map-entered variable, treat all other map-entered variables as 0's and treat all 1's as don't-cares. Find a minimal cover for the resulting map.
3. Repeat step 2 for each map-entered variable (a variable and its complement are treated as distinct variables.



(*a*) Initial map. (*b*) Map for *A*. (*c*) Map for *B*.

$$f = yz + Ay + Bx'z + B'xz + Cxy'z'$$

Heuristic Two-level Minimization

Problems with the prime implicant chart method:

- For an n-variable function, the number of prime implicants can be as large as $3^n/n$
- Prime implicant chart covering can itself be very time-consuming

Heuristic minimization: reduces the number of prime implicants that need to be tackled

ESPRESSO: expand, reduce and irredundant steps

Expand: expands implicants into prime implicants. Any implicants covered by the expanded prime implicant omitted from further consideration

Reduce: transforms prime implicants into implicants of least possible size such that all minterms of the function are still covered. This may lead to better solutions later

Irredundant: chooses a minimal subset of prime implicants obtained so far such that the subset covers all minterms of the function. Similar to prime implicant chart covering, however, less time-consuming due to fewer prime implicants

Minimization Example

Direction and order of expanding the implicant: has a bearing on the quality of the result

Example: consider circled implicant *x'y'z'*

z \ xy	00	01	11	10
0	1	1		1
1	1			1

Expansion in the *y* direction first: prime implicant *x'z'*

Expansion in the *x* direction first, then the *z* direction: *x'y'z'* -> *y'z'* -> *y'*

Expansion in the *z* direction first, then the *x* direction: *x'y'z'* -> *x'y'* -> *y'*

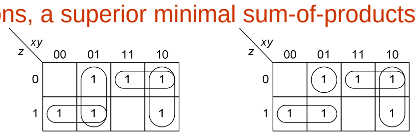
Heuristics for finding good expansion direction and order included in ESPRESSO

Example

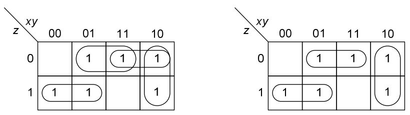
Since implicants obtained after the reduce step need not be prime, it is followed by expand and irredundant steps to derive another, possibly superior covering:

- Process continued until no longer possible to improve on the best solution seen so far
- To save time, essential prime implicants can be identified and set aside so that they are not subjected to further transformation

Example: consider an initial cover of *f* obtained by applying expand and irredundant to the initial set of minterms. After the set of transformations, a superior minimal sum-of-products is obtained.



(*a*) Initial covering of *f*. (*b*) After the reduce step.



(*c*) After the expand step. (*d*) After the irredundant step.

Equivalent Transformations Using Encoded Truth Tables

x	y	z	f
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1

expand and $\Rightarrow$ *irredundant*

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

reduce $\Rightarrow$

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

expand $\Rightarrow$

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

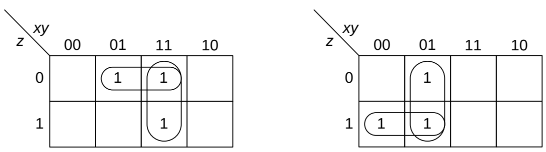
irredundant $\Rightarrow$

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

Multi-output Two-level Circuit Minimization

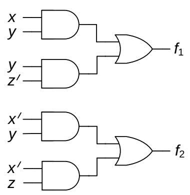
Trivial way to optimize an *n*-output circuit: minimize each circuit output separately

Example: Consider f_1 and f_2. Since all four prime implicants are essential, the two-level circuit can be obtained as shown



(a) $f_1 = xy + yz'$.

(b) $f_2 = x'y + x'z$.



(c) Two-level implementation.

Multi-output Two-level Circuit Minimization (Contd.)

Separate minimization of outputs quite sub-optimal: it does not allow sharing of logic among outputs

Multi-output prime implicant:

- Consider two functions f_1 and f_2. Their multi-output prime implicants are the prime implicants of f_1, f_2 and f_1f_2.
- For three functions, f_1, f_2, and f_3, the multi-output prime implicants are those of f_1, f_2, f_3, f_1f_2, f_1f_3, f_2f_3, and $f_1f_2f_3$.
- Similarly, for n outputs, the number of functions to be considered is 2^n-1.

Scenario - prime implicant of f_1 is also a prime implicant of f_1f_2: then consider this prime implicant for only f_1f_2, not for f_1. This enables the sharing of the prime implicant among more outputs.

Augmented Prime Implicant Chart

Augmented chart: has rows corresponding to each of the 2^n-1 functions that has at least one prime implicant that deserves further consideration and columns corresponding to the minterms of the function

- To minimize the number of gates – usual steps: identify essential prime implicants, remove dominated rows and dominating columns, using branch-and-bound or prime implicant function when necessary
- If secondary objective is to minimize the number of interconnects: removing dominated rows is not allowed since this sometimes eliminates the solution with fewer interconnects

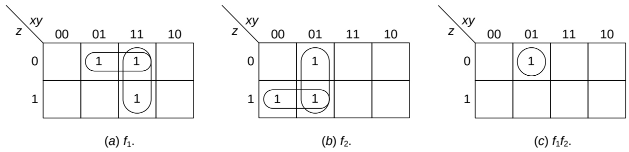
Augmented Chart Example

Example: Consider f_1 and f_2

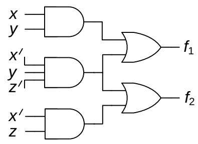
- Since none of the prime implicants of f_1 and f_2 are also a prime implicant of f_1f_2, all five prime implicants deserve further consideration
- Since dominated rows cannot be eliminated, we use prime implicant function p on the reduced chart

$$p = (B+E)(C+E) = BC + E$$

- This leads to the multi-output implementation



(a) f_1. (b) f_2. (c) f_1f_2.



Function	Prime implicant	f_1: 2	f_1: ✓6	f_1: ✓7	f_2: ✓1	f_2: 2	f_2: ✓3
f_1	✓A = xy		×	×			
	B = yz'	×	×				
f_2	C = $x'y$					×	×
	✓D = $x'z$				×		×
f_1f_2	E = $x'yz'$	×				×	

(d) Augmented prime implicant chart.

Function	Prime implicant	f_1 2	f_2 2
f_1	B = yz'	×	
f_2	C = $x'y$		×
f_1f_2	E = $x'yz'$	×	×

(e) Reduced chart.

41

Equivalent Transformations Using Encoded Truth Tables

x	y	z	f_1	f_2
--	1	0	1	0
1	1	--	1	0
0	--	1	0	1
0	1	--	0	1

reduce ⇒

x	y	z	f_1	f_2
--	1	0	1	0
1	1	--	1	0
0	--	1	0	1
0	1	--	0	1
0	1	0	1	1

irredundant ⇒

x	y	z	f_1	f_2
1	1	--	1	0
0	--	1	0	1
0	1	0	1	1

42