McGill

**ECSE 323**
**Digital System Design**

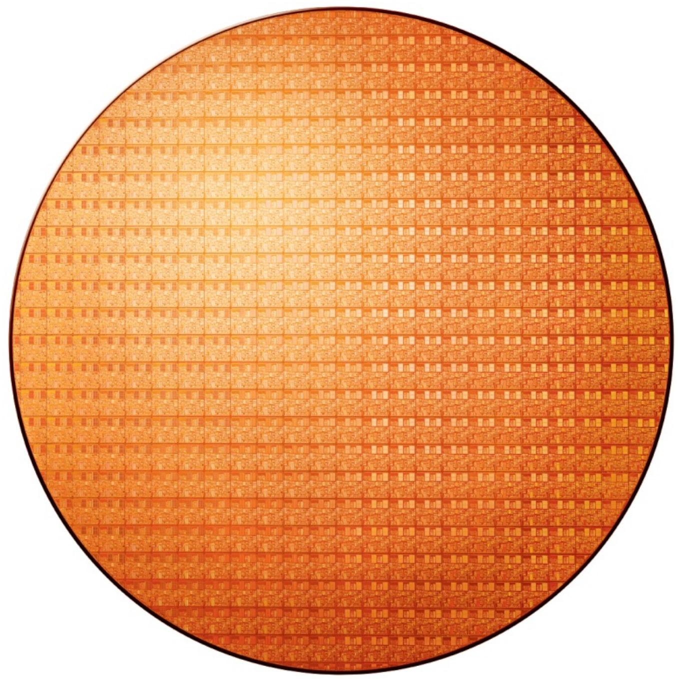
# Logic Minimization

**Prof. Warren Gross**

Material used in this set of slides was based on "*Fundamentals of Digital Logic with VHDL Design*"
by S. Brown and Z. Vranesic

# Why?

- Why do we want to reduce the *cost* of our digital circuits?



Silicon wafer

# Logic Minimization

- This topic covers parts of Chapter 4 of the textbook (4.1 – 4.5, 4.8, 4.9, 4.11)
- We will come back and cover 4.6 and 4.7 when we look at combinational circuits

# Logic Minimization

- It is easy to find the canonical SOP or POS form of a logic function
- Finding minimal realizations not so straightforward
  - Boolean algebra transformation can be tedious and are impractical for functions with a large number of input variables
- We will study two optimization techniques

# Karnaugh Map

- Key idea in minimization is to reduce the number of product (or sum) terms, and the number of variables in each term.
- Karnaugh map is a graphical technique that makes it easy to use the combining property:

# Two-Variable Map

| $x_1$ | $x_2$ | |
|---|---|---|
| 0 | 0 | $m_0$ |
| 0 | 1 | $m_1$ |
| 1 | 0 | $m_2$ |
| 1 | 1 | $m_3$ |

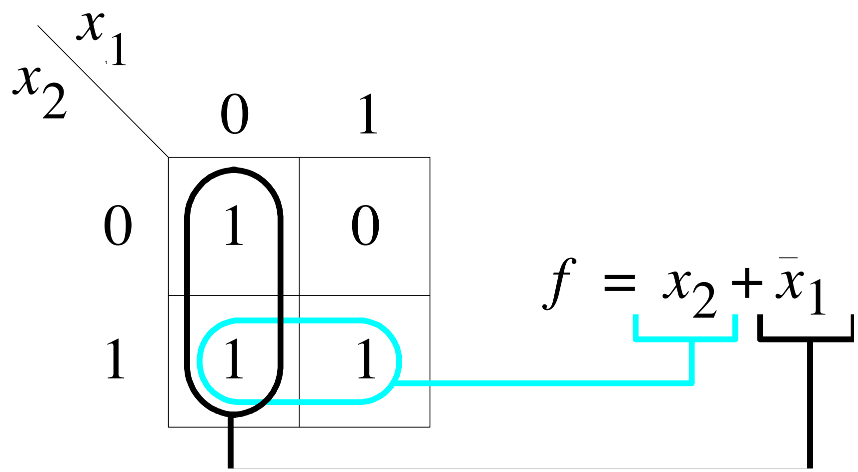
(a) Truth table

| $x_2$ \ $x_1$ | 0 | 1 |
|---|---|---|
| 0 | $m_0$ | $m_2$ |
| 1 | $m_1$ | $m_3$ |

(b) Karnaugh map

| $x_1$ | $x_2$ | $f(x_1, x_2)$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

$f = x_1'x_2' + x_1'x_2 + x_1x_2$




- If a variable is included in all its possible values in a grouping, then it can be removed

- Find the smallest number of product terms that *cover* the function
    - Draw the largest possible rectangular groupings
    - cost of each of the product terms should be as low as possible

- Two variable map → can have groups of 1, 2 or 4 minterms
    - group of 1: 2 *literals* in the product term
    - group of 2: 1 literal
    - group of 4: 0 literals (the trivial function *f* = 1)

# Three-Variable Map

| $x_1$ | $x_2$ | $x_3$ | |
|---|---|---|---|
| 0 | 0 | 0 | $m_0$ |
| 0 | 0 | 1 | $m_1$ |
| 0 | 1 | 0 | $m_2$ |
| 0 | 1 | 1 | $m_3$ |
| 1 | 0 | 0 | $m_4$ |
| 1 | 0 | 1 | $m_5$ |
| 1 | 1 | 0 | $m_6$ |
| 1 | 1 | 1 | $m_7$ |

(a) Truth table

| $x_3$ \ $x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $m_0$ | $m_2$ | $m_6$ | $m_4$ |
| 1 | $m_1$ | $m_3$ | $m_7$ | $m_5$ |

(b) Karnaugh map

- Label with Gray Code
- Columns 1 and 4 differ only in $x_1$ → adjacent
- Groups of 1, 2, 4, (8)

| $x_3$ \ $x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |

$$f = x_1\bar{x}_3 + \bar{x}_2x_3$$

| $x_3$ \ $x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |

$$f = \bar{x}_3 + x_1\bar{x}_2$$

# Four-Variable Map

| $x_3x_4 \backslash x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | $m_0$ | $m_4$ | $m_{12}$ | $m_8$ |
| 01 | $m_1$ | $m_5$ | $m_{13}$ | $m_9$ |
| 11 | $m_3$ | $m_7$ | $m_{15}$ | $m_{11}$ |
| 10 | $m_2$ | $m_6$ | $m_{14}$ | $m_{10}$ |

$x_1$ $x_2$ $x_3$ $x_4$

- Groups of 1, 2, 4, 8, (16)
- 4, 3, 2, 1, (0) literals

| $x_3x_4 \backslash x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 1 | 1 |
| 11 | 1 | 0 | 0 | 1 |
| 10 | 1 | 0 | 0 | 1 |

$$f_1 = \bar{x}_2 x_3 + x_1 \bar{x}_3 x_4$$

| $x_3x_4 \backslash x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 1 | 1 |
| 11 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 |

$$f_2 = x_3 + x_1 x_4$$

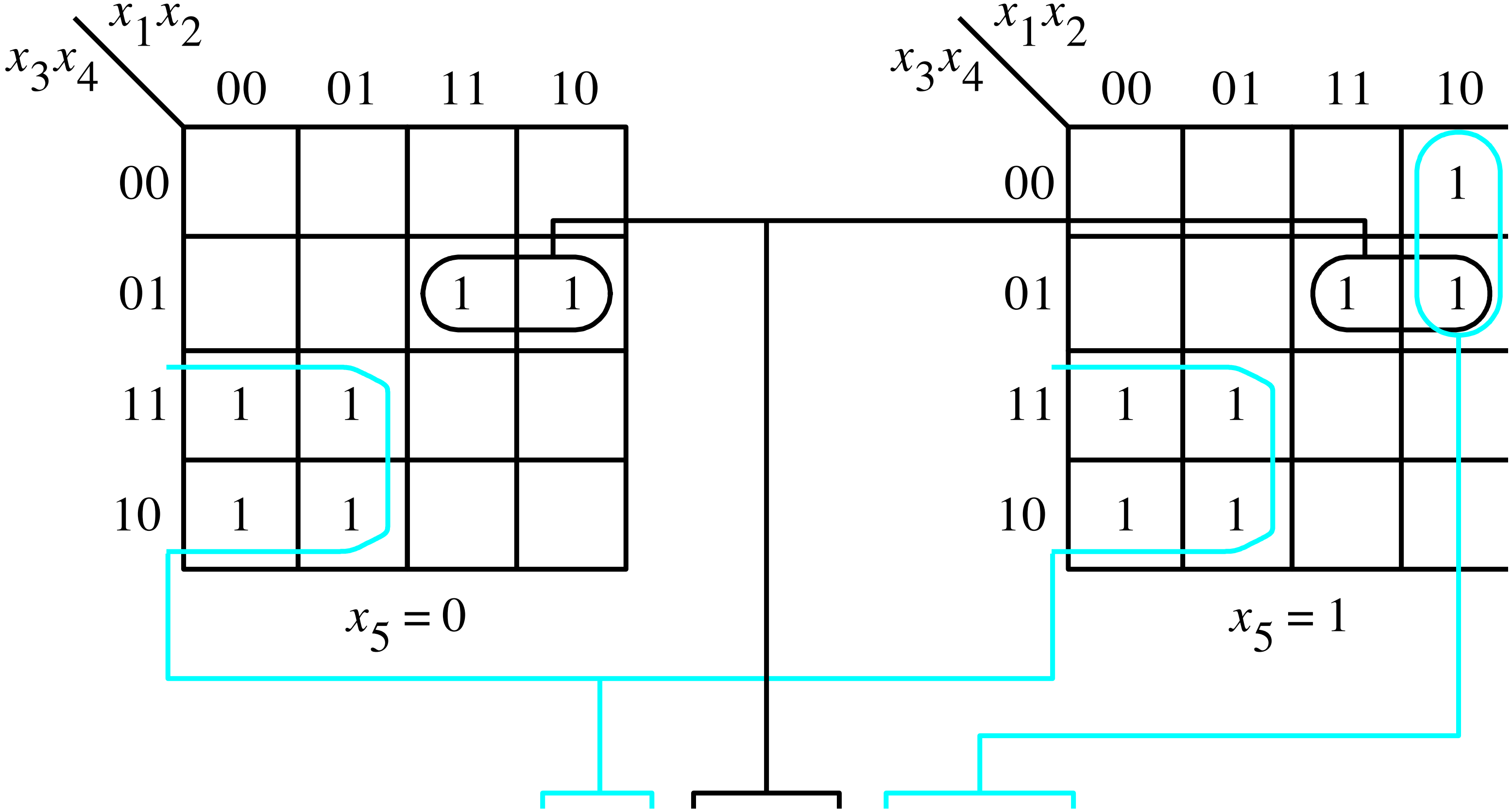
| $x_3x_4 \backslash x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 0 | 1 |
| 01 | 0 | 0 | 0 | 0 |
| 11 | 1 | 1 | 1 | 0 |
| 10 | 1 | 1 | 0 | 1 |

$$f_3 = \bar{x}_2 \bar{x}_4 + \bar{x}_1 x_3 + x_2 x_3 x_4$$

| $x_3x_4 \backslash x_1x_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | 1 | 0 |
| 01 | 1 | 1 | 1 | 0 |
| 11 | 0 | 0 | 1 | 1 |
| 10 | 0 | 0 | 1 | 1 |

$f_4 = \bar{x}_1 \bar{x}_3 + x_1 x_3 + x_1 x_2$ or $x_2 \bar{x}_3$

# Five-Variable Map

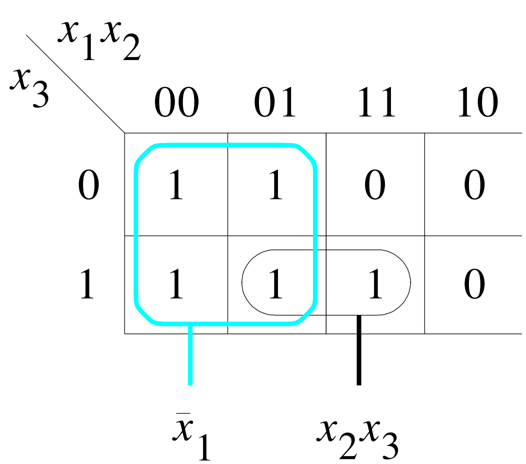



$$f_1 = \bar{x}_1 x_3 + x_1 \bar{x}_3 x_4 + x_1 \bar{x}_2 \bar{x}_3 x_5$$

# Minimization Strategy Terminology

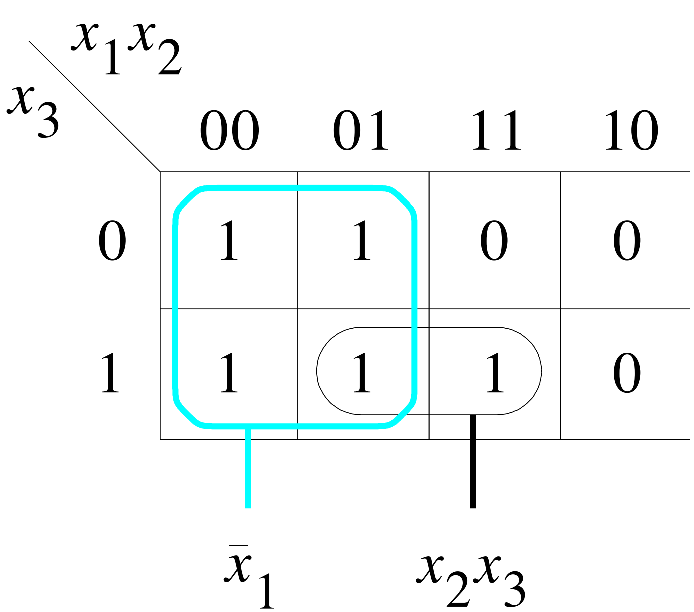
- ***Literal:*** Each appearance of a variable either in true or complemented form
  - e.g. $x_1x_2'x_3$ has 3 literals
  - $x_1'x_3x_4'x_6$ has 4 literals

- ***Implicant:*** A product term that indicates the input valuation(s) for which a given function *f* is equal to 1 is called an implicant of *f*.
  - Example : the minterms are the most basic implicants (that consist of $n$ literals)



# Terminology

- ***Prime Implicant:*** An implicant is called a *prime implicant* if it cannot be combined into another implicant that has fewer literals.
  - i.e. if you delete a literal in a prime implicant, it is no longer a valid implicant
  - Karnaugh map is one way to find the prime implicants (the largest groupings)



# Terminology

- A ***cover*** is a collection of implicants that account for all valuations for which a given function is equal to 1.

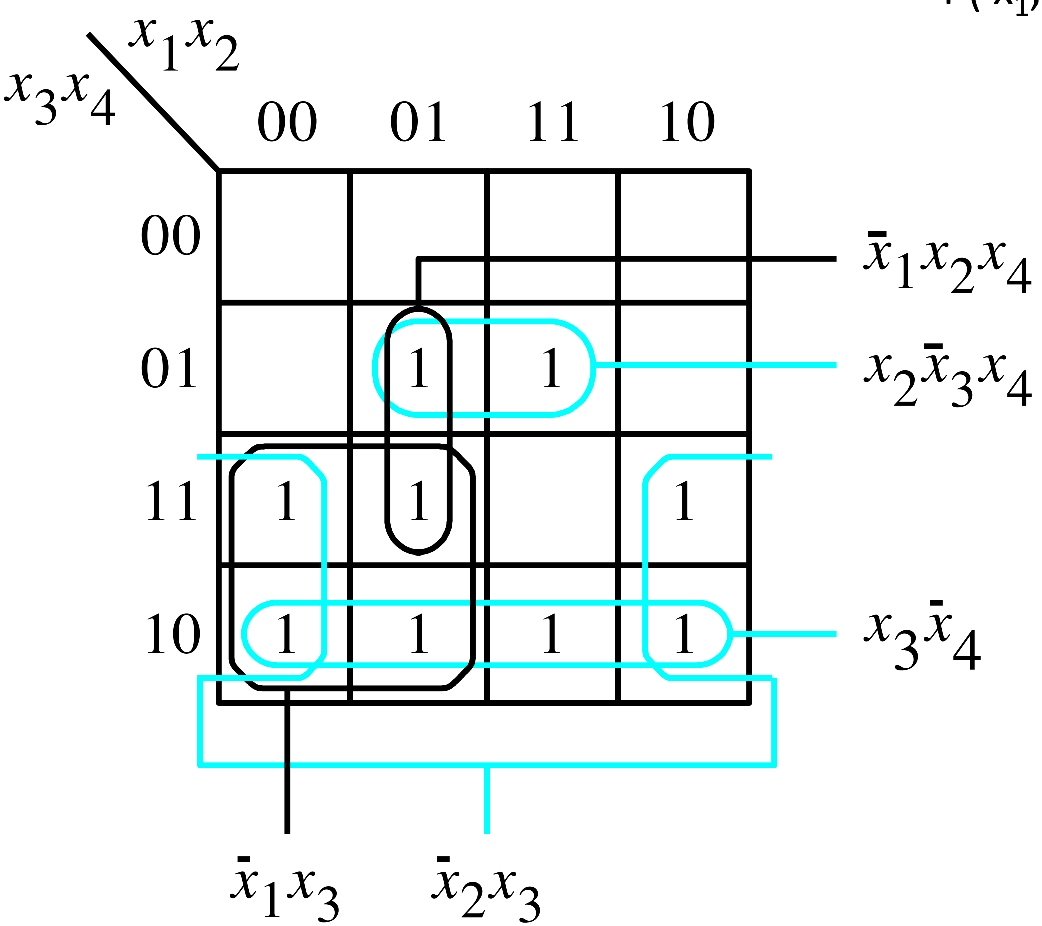
- Most functions have a number of different covers
  - E.g. Set of all minterms for which $f = 1$ is a cover
  - E.g. Set of all prime implicants is a cover

- Want to choose a cover that has the lowest ***cost***
  - # gates + total # of inputs to all gates
  - Assume primary inputs are available in both true and complemented form at zero cost
  - If an inversion is needed inside the circuit, the corresponding NOT gate and its input is included in the cost

# Minimization Procedure

- Find the minimum-cost subset of the prime implicants that will cover the function
  - Some may have to be included (***essential prime implicants***) while for others there may be a choice
  - To choose which of the non-essential prime implicants to include in the cover we usually use a heuristic
    - A heuristic does not consider ALL the possibilities, but results in a good result most of the time
    - CAD tools make use of many heuristics

$f(x_1, \ldots, x_4) = \Sigma\, m(2, 3, 5, 6, 7, 10, 11, 13, 14)$




- 5 prime implicants
- Essential prime implicants (include a minterm only covered by one PI)
- $m_7$ not covered by the essential prime implicants
- $m_7$ can be covered by either $x_1'x_3$ or $x_1'x_2x_4$. Choose $x_1'x_3$ because of lower cost.

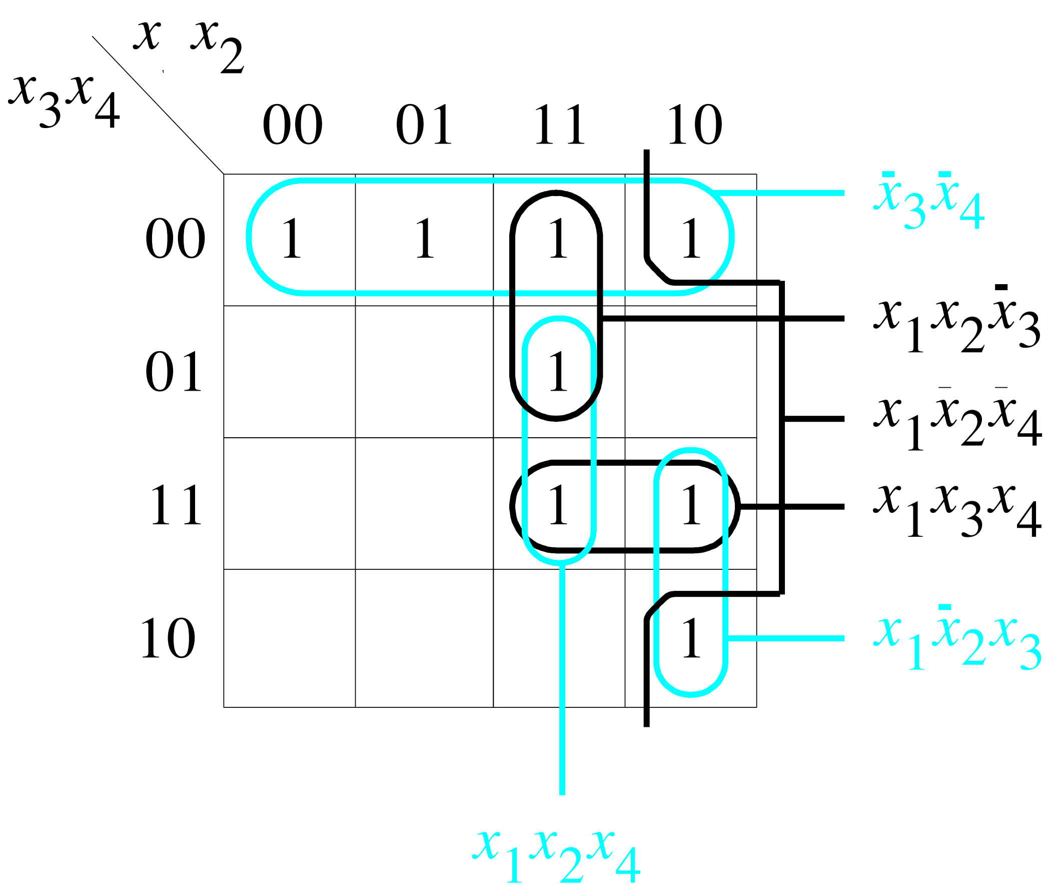
# Minimization Procedure

1. Generate the set of prime implicants for $f$
2. Find the set of essential prime implicants
3. If the set of essential prime implicants covers $f$ then choose this set. Otherwise determine the nonessential prime implicants that should be added to for a complete minimum-cost cover

- A heuristic: select one non-essential prime implicant in the cover and find the rest of the cover.
- Next, assume it is not in the cover and find another cover. Choose the cover with the lowest cost.

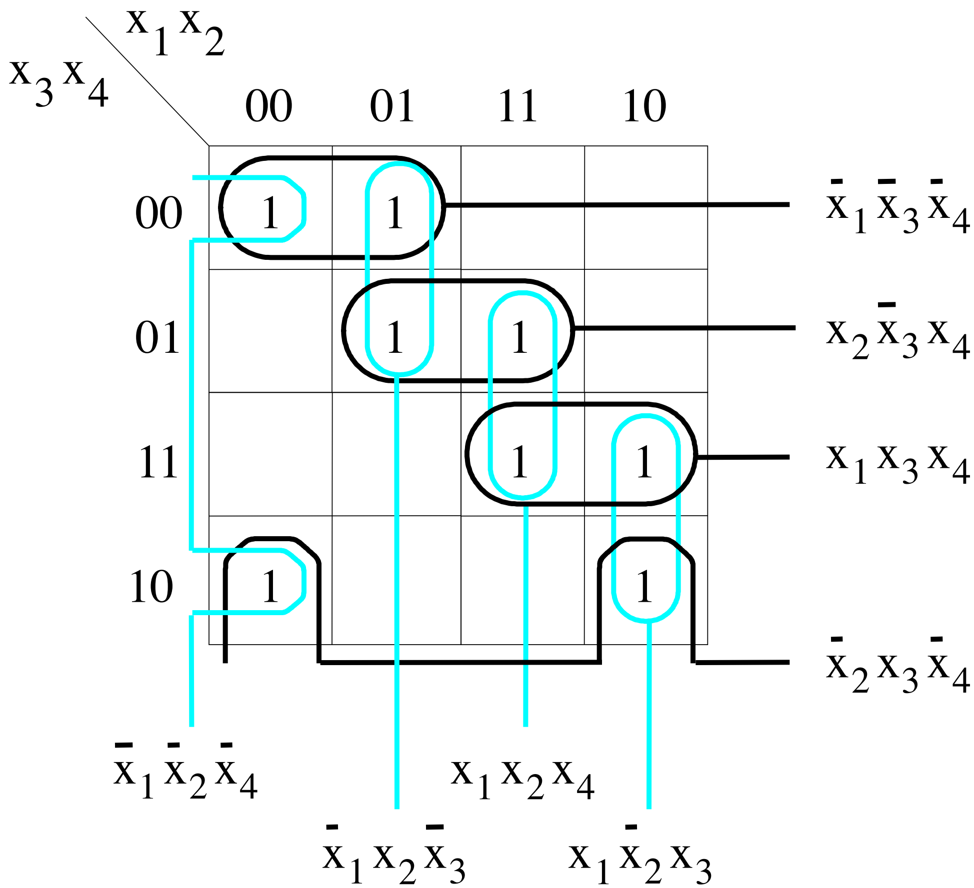
$f\ (x_1,\ldots,x_4) = \Sigma\, m(0, 4, 8, 10, 11, 12, 13, 15)$

Only $x_3'x_4'$ is essential – include in cover




One choice (include $x_1x_2x_3'$): $f = x_3'x_4' + x_1x_2x_3' + x_1x_3x_4 + x_1x_2'x_3$

Another choice (do not include $x_1x_2x_3'$): $f = x_3'x_4' + x_1x_2x_4 + x_1x_2'x_3$ (lower cost)

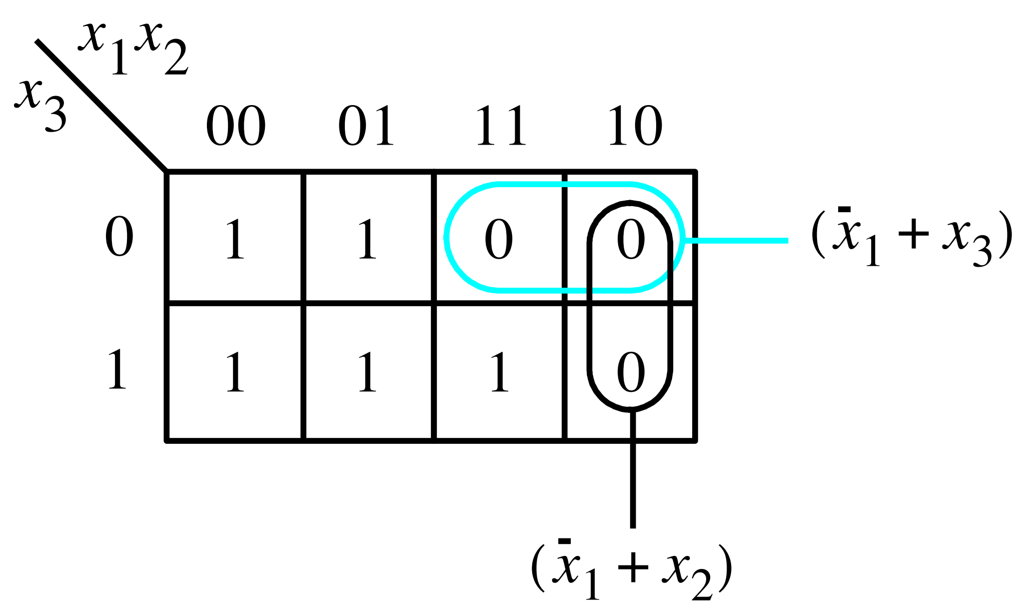



- $f(x_1,\ldots, x_4) = \Sigma\, m(0, 2, 4, 5, 10, 11, 13, 15)$
- All prime implicants are non-essential
- Choose any prime implicant and then exclude it – both alternatives (blue and black sets of prime implicants) are of equal cost in this example

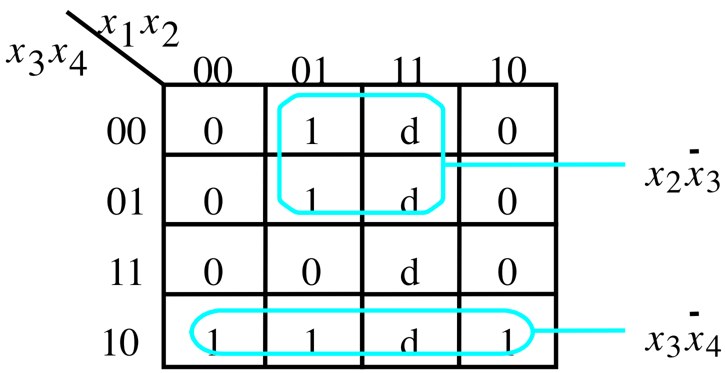
# Minimization of POS Forms

- CAD tools usually synthesize both the SOP and POS forms and select the one with the lowest cost

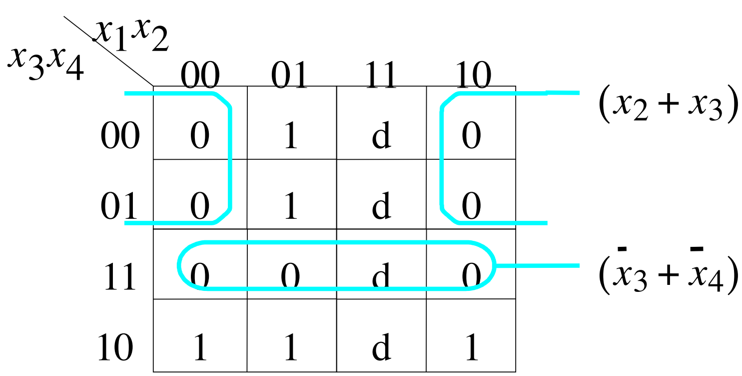
$f(x_1, x_2, x_3) = \Pi\, M(4, 5, 6)$
$= (x_1' + x_2)(x_1' + x_3)$



# Incompletely Specified Functions




(a) SOP implementation




(b) POS implementation

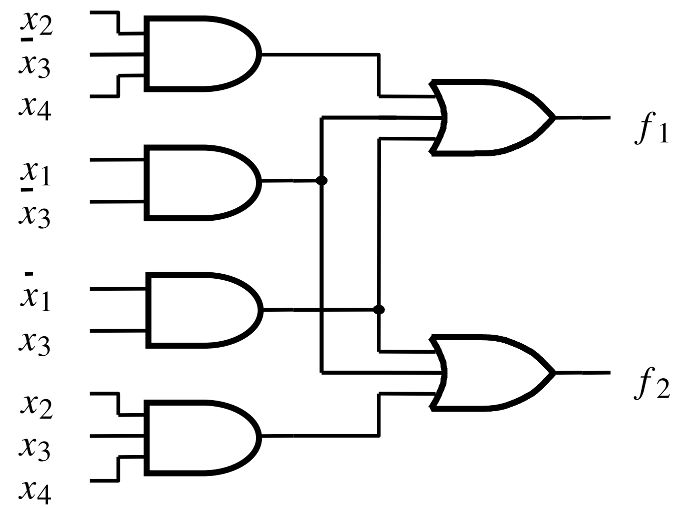
- Certain input combinations many be known to never happen
  - "Don't-care" condition
- Can assign the function value to give the lowest cost circuit
  - (can make different choices for the SOP and POS forms)

$f(x_1, \ldots, x_4) = \Sigma\, m(2, 4, 5, 6, 10) + D(12, 13, 14, 15)$

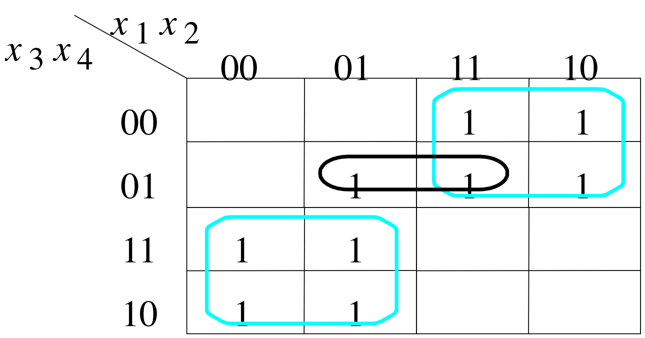
# Multiple-Output Circuits

- Sometimes, circuits share some of the same logic gates and can be combined

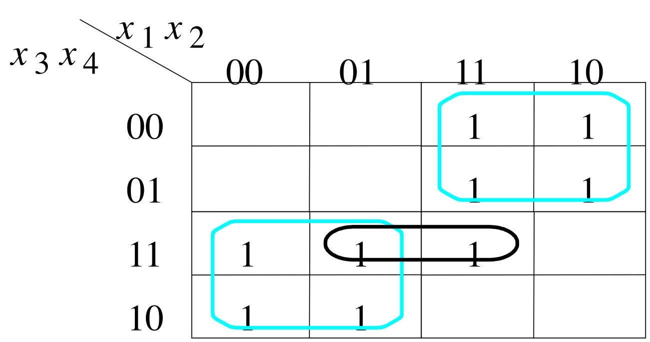
- One strategy: find minimum-cost realization of $f_1$ and $f_2$ and the share the common product terms



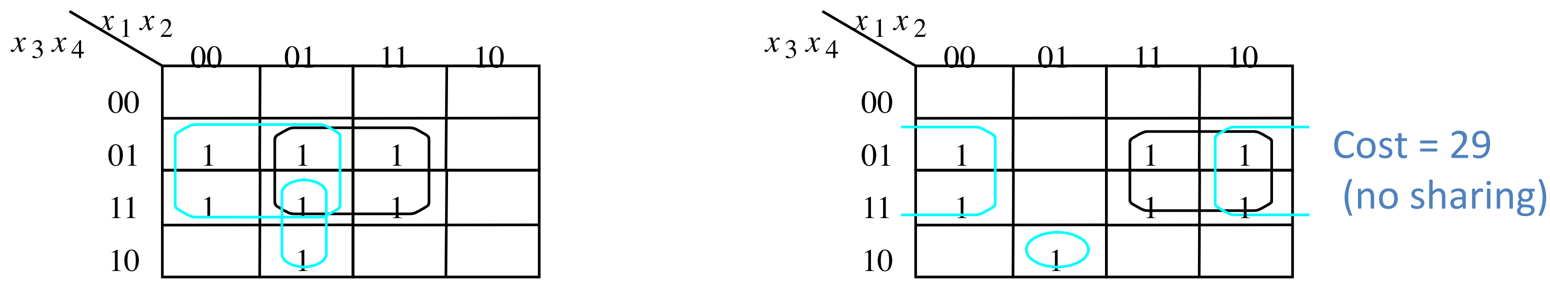

(c) Combined circuit for $f_1$ and $f_2$




(a) Function $f_1$

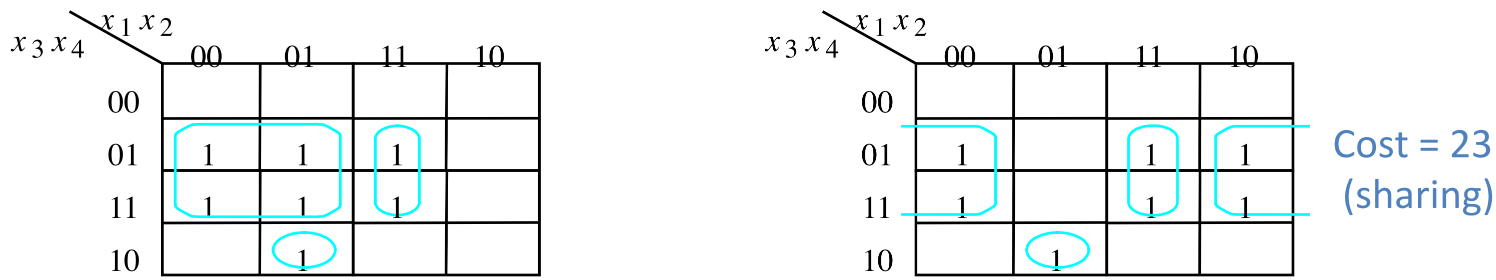



(b) Function $f_2$

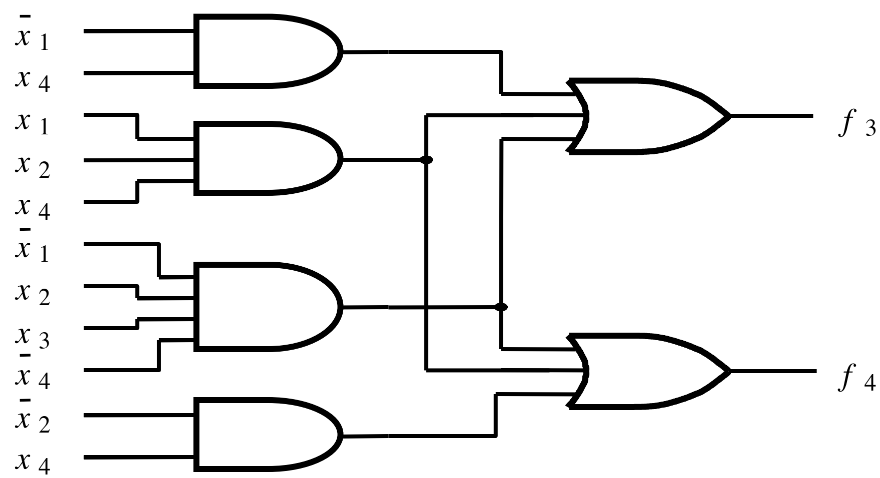



(a) Optimal realization of $f_3$

(b) Optimal realization of $f_4$




(c) Optimal realization of $f_3$ and $f_4$ together




(d) Combined circuit for $f_3$ and $f_4$

Another strategy: finding optimal realizations of $f_3$ and $f_4$ independently is not a good idea in this case

# Cubical Representation

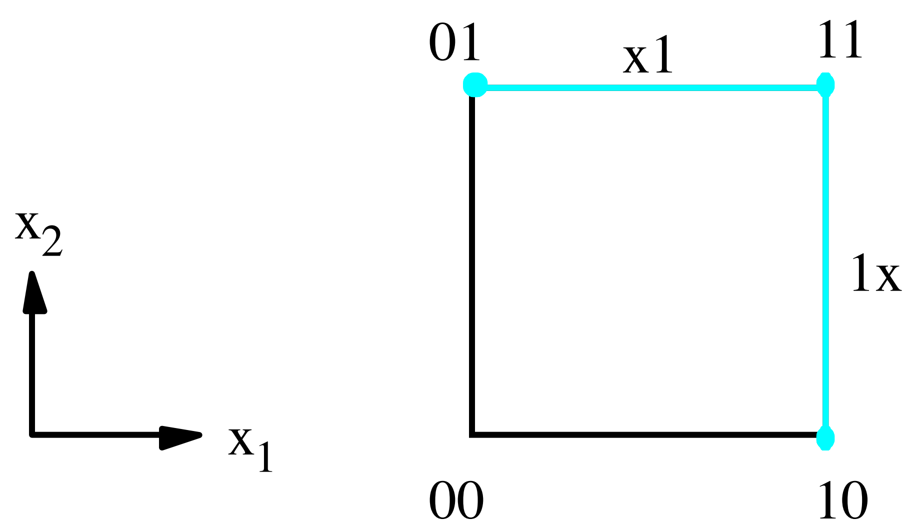
- Karnaugh map is only useful for minimizing functions of a small number of variables
- We will study more a systematic algebraic optimization technique
- There are many techniques based on ***cubical representation***
  - Truth table
  - Algebraic expression
  - Venn Diagram
  - Karnaugh map
  - Cubical representation
- Cubical representation is used in modern CAD tools

e.g. SYNOPSYS® ALTERA. QUARTUS® II

# 2-Cube

- Map a function of *n* variables onto an *n*-dimensional cube (*n*-cube)
- An edge joins two vertices for which the labels differ in the value of only one variable
- Finding edges for which *f* = 1 is equivalent to applying the combining property (14a)
  - Same as finding pairs of adjacent cells for which *f* = 1 in a Karnaugh map

- $f = \{1x, x1\} \rightarrow f = x_1 + x_2$



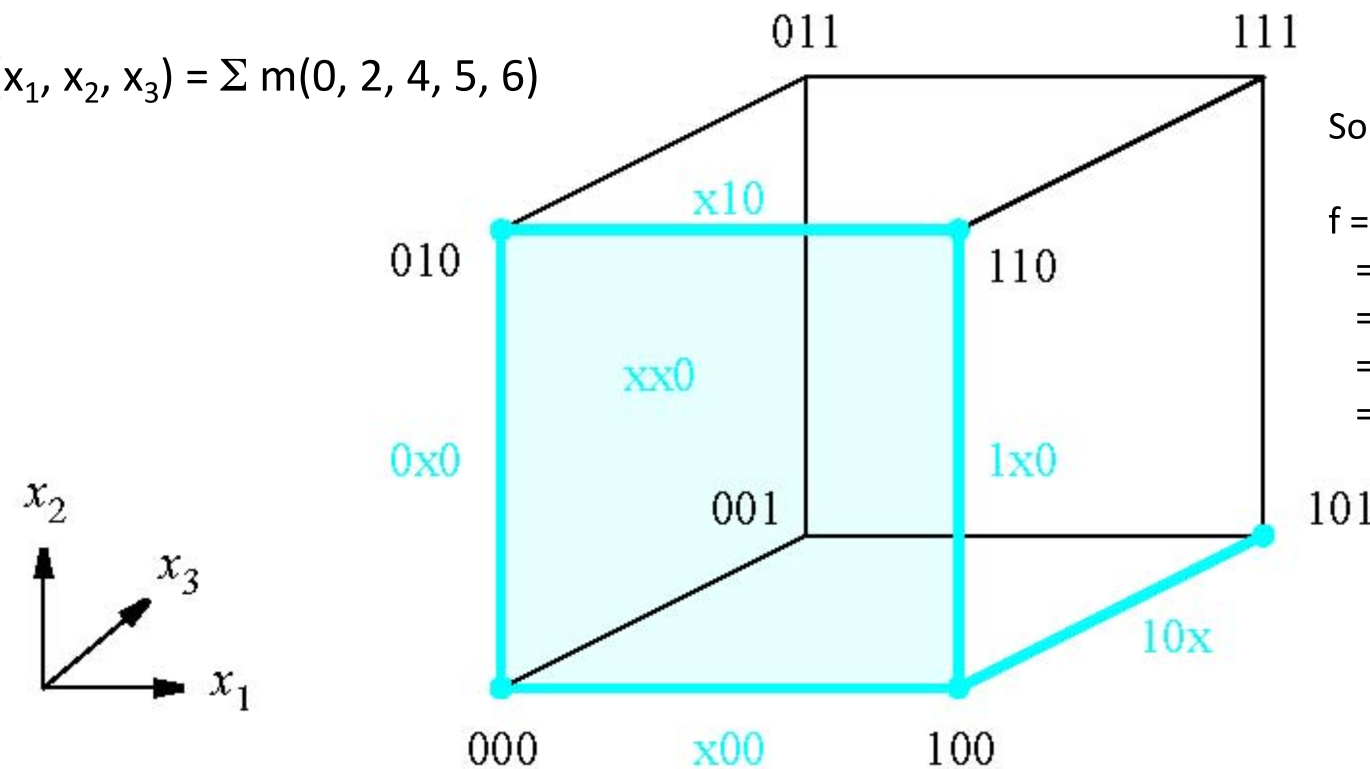

| $x_1$ | $x_2$ | f |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

$f(x_1, x_2) = \Sigma\, m(1, 2, 3)$.

# 3-Cube

$f = \{xx0, 10x\} = x_3' + x_1x_2'$ is least-expensive

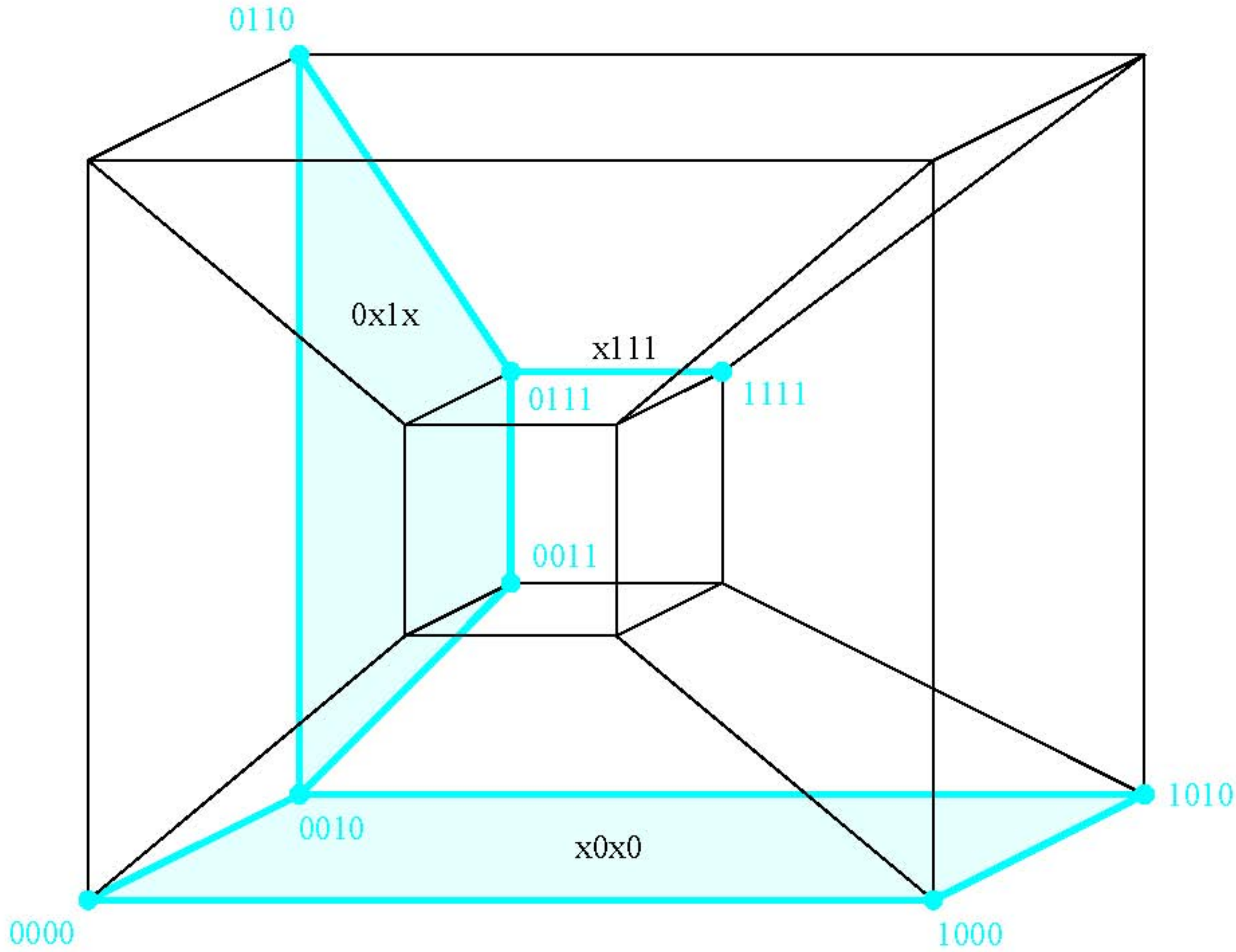
$f\ (x_1, x_2, x_3) = \Sigma\ m(0, 2, 4, 5, 6)$




Some possible representations:

f = {000,010,100,101,110}
= {0x0, 1x0, 101}
= {x00,x10,101}
= {x00,x10,10x}
= {xx0,10x}

# 4-Cube



# *n*-cube

- It is impractical to draw a cube of more than 4-dimensions, but it is easy to extend the concept to *n*-dimensions.
- *n*-cubes contain lower dimensional cubes (which contain lower dimensional cubes…)
- $2^k$ adjacent vertices – *k*-cube
  - Vertex: 0-cube
  - Edge: 1-cube
  - Side: 2-cube
  - etc…
- The largest possible *k*-cubes that exist for a given function are its prime implicants !

# Quine-McCluskey Method

- A tabular method based on cubical representation
- The tabular form is more suited to implementation in a computer algorithm
- Two-steps
    1. Generation of Prime Implicants
    2. Determination of a Minimum Cover

# Generation of Prime Implicants

- Prime Implicants are the largest possible *k*-cubes for which either $f = 1$ or $f$ is unspecified (don't-care)
- Assume $f$ and it's don't-cares are specified as minterms
  – Write down the vertices of $f$
- Key idea: compare vertices in pairwise fashion to see if they can be combined into larger cubes
  – Then see if these cubes can be combined into larger cubes, etc…keep going until you find the prime implicants
- Two cubes that are identical in all variables (coordinates) except one, for which one cube has the value 0 and the other has 1, then these cubes can be combined into a larger cube.

# Example 1

- $f(x_1,x_2,x_3,x_4) = \{1000,1001,1010,1011\}$
- Cubes 1000 and 1001 differ only in one variable ($x_4$)
  - Combine into new cube 100x
- Cubes 1010 and 1011 can be combined into 101x
- Combine 100x and 101x into 10xx
- $f = x_1x_2'$

# Example 2

- Group the minterms such that the cubes in each group have the same number of 1s and sort the groups by the number of 1s
- Only need to compare each cube in a group with all the cubes in the immediately preceding group
- Place a check if a ($k$-1)-cube is included in a $k$ cube

List 1 (0-cubes)

| | | |
|---|---|---|
| 0 | 0 0 0 0 | ✓ |
| 4<br>8 | 0 1 0 0<br>1 0 0 0 | ✓<br>✓ |
| 10<br>12 | 1 0 1 0<br>1 1 0 0 | ✓<br>✓ |
| 11<br>13 | 1 0 1 1<br>1 1 0 1 | ✓<br>✓ |
| 15 | 1 1 1 1 | ✓ |

List 2 (1-cubes)

| | | |
|---|---|---|
| 0,4<br>0,8 | 0 x 0 0<br>x 0 0 0 | ✓<br>✓ |
| 8,10<br>4,12<br>8,12 | 1 0 x 0<br>x 1 0 0<br>1 x 0 0 | <br>✓<br>✓ |
| 10,11<br>12,13 | 1 0 1 x<br>1 1 0 x | |
| 11,15<br>13,15 | 1 x 1 1<br>1 1 x 1 | |

List 3 (2-cubes)

| | |
|---|---|
| 0,4,8,12 | x x 0 0 |

The cubes without a check are the prime implicants of $f$

P = {10x0, 101x, 110x, 1x11, 11x1, xx00}
= $\{p_1,p_2,p_3,p_4,p_5,p_6\}$

$f(x_1,\ldots,x_4) = \Sigma\, m(0,4,8,10,11,12,13,15)$

# Determination of a Minimum Cover

- Find a minimum-cost cover for *f*
- Assume that the cost is directly proportional to the number of inputs to all gates
  - i.e. the number of literals in the prime implicants
- Construct a *prime-implicant cover table*
  - A row for each prime implicant
  - A column for each minterm to be covered
  - Put checks to indicate the minterms covered by each prime implicant
- If there is a single check in some column, then the prime implicant that covers the minterm of this column is *essential* and it must be included in the final cover
  - Remove the rows corresponding to the essential prime implicants and the columns of the minterms associated with them

| Prime implicant | Minterm 0 | 4 | 8 | 10 | 11 | 12 | 13 | 15 |
|---|---|---|---|---|---|---|---|---|
| $p_1$ = 1 0 x 0 | | | ✓ | ✓ | | | | |
| $p_2$ = 1 0 1 x | | | | ✓ | ✓ | | | |
| $p_3$ = 1 1 0 x | | | | | | ✓ | ✓ | |
| $p_4$ = 1 x 1 1 | | | | | ✓ | | | ✓ |
| $p_5$ = 1 1 x 1 | | | | | | | ✓ | ✓ |
| $p_6$ = x x 0 0 | ✓ | ✓ | ✓ | | | ✓ | | |

Initial prime implicant cover table

| Prime implicant | Minterm 10 | 11 | 13 | 15 |
|---|---|---|---|---|
| $p_1$ | ✓ | | | |
| $p_2$ | ✓ | ✓ | | |
| $p_3$ | | | ✓ | |
| $p_4$ | | ✓ | | ✓ |
| $p_5$ | | | ✓ | ✓ |

After the removal of essential prime implicants

# Row-Dominance

- $p_1$ only covers minterm 10 while $p_2$ covers both minterm 10 and minterm 11
  - $p_2$ *dominates* $p_1$
  - Cost of $p_2$ is the same as $p_1$ therefore choose $p_2$ rather than $p_1$ and remove $p_1$ from the table
- $p_5$ dominates $p_3$
  - Remove $p_3$
- Must choose $p_2$ to cover minterm 10 and $p_5$ to cover minterm 13 which takes care of covering minterm 11 and minterm 15
- Final cover is C = {$p_2$,$p_5$,$p_6$} = {101x, 11x1,xx00}
- $f = x_1x_2'x_3 + x_1x_2x_4 + x_3'x_4'$
- Do not remove a dominated row if the cost of its prime implicant is less than that of the dominating row's prime implicant

| Prime implicant | Minterm 10 | 11 | 13 | 15 |
|---|---|---|---|---|
| $p_1$ | ✓ | | | |
| $p_2$ | ✓ | ✓ | | |
| $p_3$ | | | ✓ | |
| $p_4$ | | ✓ | | ✓ |
| $p_5$ | | | ✓ | ✓ |

| Prime implicant | Minterm 10 | 11 | 13 | 15 |
|---|---|---|---|---|
| $p_2$ | ✓ | ✓ | | |
| $p_4$ | | ✓ | | ✓ |
| $p_5$ | | | ✓ | ✓ |

After the removal of dominated rows

# Example 3 – Column Dominance

- $f(x_1,\ldots,x_4) = \Sigma\, m(0,2,5,6,7,8,9,13) + D(1,12,15)$
- P = {00x0,0x10,011x,x00x,xx01,1x0x,x1x1} = $\{p_1,p_2,p_3,p_4,p_5,p_6,p_7\}$
- Do not include don't care minterms in the prime implicant cover table since they don't need to be covered

List 1

| | | |
|---|---|---|
| 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 | ✓ |
| 6 | 0 1 1 0 | ✓ |
| 9 | 1 0 0 1 | ✓ |
| 12 | 1 1 0 0 | ✓ |
| 7 | 0 1 1 1 | ✓ |
| 13 | 1 1 0 1 | ✓ |
| 15 | 1 1 1 1 | ✓ |

List 2

| | | |
|---|---|---|
| 0,1 | 0 0 0 x | ✓ |
| 0,2 | 0 0 x 0 | |
| 0,8 | x 0 0 0 | ✓ |
| 1,5 | 0 x 0 1 | ✓ |
| 2,6 | 0 x 1 0 | |
| 1,9 | x 0 0 1 | ✓ |
| 8,9 | 1 0 0 x | ✓ |
| 8,12 | 1 x 0 0 | ✓ |
| 5,7 | 0 1 x 1 | ✓ |
| 6,7 | 0 1 1 x | |
| 5,13 | x 1 0 1 | ✓ |
| 9,13 | 1 x 0 1 | ✓ |
| 12,13 | 1 1 0 x | ✓ |
| 7,15 | x 1 1 1 | ✓ |
| 13,15 | 1 1 x 1 | ✓ |

List 3

| | |
|---|---|
| 0,1,8,9 | x 0 0 x |
| 1,5,9,13 | x x 0 1 |
| 8,9,12,13 | 1 x 0 x |
| 5,7,13,15 | x 1 x 1 |

| Prime implicant | | Minterm 0 | 2 | 5 | 6 | 7 | 8 | 9 | 13 |
|---|---|---|---|---|---|---|---|---|---|
| $p_1$ | = 0 0 x 0 | ✓ | ✓ | | | | | | |
| $p_2$ | = 0 x 1 0 | | ✓ | | ✓ | | | | |
| $p_3$ | = 0 1 1 x | | | | ✓ | ✓ | | | |
| $p_4$ | = x 0 0 x | ✓ | | | | | ✓ | ✓ | |
| $p_5$ | = x x 0 1 | | | ✓ | | | | ✓ | ✓ |
| $p_6$ | = 1 x 0 x | | | | | | ✓ | ✓ | ✓ |
| $p_7$ | = x 1 x 1 | | | ✓ | | ✓ | | | ✓ |

(a) Initial prime implicant cover table

- Column 9 dominates column 8
- Remove the **dominating** column (9)
- Column 13 dominates column 5
- Remove column 13

| Prime implicant | | Minterm 0 | 2 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| $p_1$ | = 0 0 x 0 | ✓ | ✓ | | | | |
| $p_2$ | = 0 x 1 0 | | ✓ | | ✓ | | |
| $p_3$ | = 0 1 1 x | | | | ✓ | ✓ | |
| $p_4$ | = x 0 0 x | ✓ | | | | | ✓ |
| $p_5$ | = x x 0 1 | | | ✓ | | | |
| $p_6$ | = 1 x 0 x | | | | | | ✓ |
| $p_7$ | = x 1 x 1 | | | ✓ | | ✓ | |

(b) After the removal of columns 9 and 13

Row $p_4$ dominates $p_6$ and $p_7$ dominates $p_5$

| Prime implicant | Minterm 0 | 2 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|
| $p_1$ | ✓ | ✓ | | | | |
| $p_2$ | | ✓ | | ✓ | | |
| $p_3$ | | | | ✓ | ✓ | |
| $p_4$ | ✓ | | | | | ✓ |
| $p_7$ | | | ✓ | | ✓ | |

(c) After the removal of rows $p_5$ and $p_6$

| Prime implicant | Minterm 2 | 6 |
|---|---|---|
| $p_1$ | ✓ | |
| $p_2$ | ✓ | ✓ |
| $p_3$ | | ✓ |

(d) After including $p_4$ and $p_7$ in the cover

$C = \{p_2, p_4, p_7\}$
$= \{0x10, x00x, x1x1\}$

$$f = x_1'x_3x_4' + x_2'x_3' + x_2x_4$$

# Example 4

- No essential prime implicants or dominating rows or columns
- All prime implicants have equal cost
- Use *branching*
  - Choose any prime implicant, say p3 and include it in cover, compute rest of cover and its cost
  - $C_1 = \{p_1, p_3, p_4\}$
  - Now, find a new cover excluding p3 and compare costs
  - $C_2 = \{p_1, p_5\} = C_{min}$

| Prime implicant | Minterm 0 | 3 | 10 | 15 |
|---|---|---|---|---|
| $p_1$ = 0 0 x x | ✓ | ✓ | | |
| $p_2$ = x 0 x 0 | ✓ | | ✓ | |
| $p_3$ = x 0 1 x | | ✓ | ✓ | |
| $p_4$ = x x 1 1 | | ✓ | | ✓ |
| $p_5$ = 1 x 1 x | | | ✓ | ✓ |

(a) Initial prime implicant cover table

| Prime implicant | Minterm 0 | 15 |
|---|---|---|
| $p_1$ | ✓ | |
| $p_2$ | ✓ | |
| $p_4$ | | ✓ |
| $p_5$ | | ✓ |

(b) After including $p_3$ in the cover

| Prime implicant | Minterm 0 | 3 | 10 | 15 |
|---|---|---|---|---|
| $p_1$ | ✓ | ✓ | | |
| $p_2$ | ✓ | | ✓ | |
| $p_4$ | | ✓ | | ✓ |
| $p_5$ | | | ✓ | ✓ |

(c) After excluding $p_3$ from the cover

# Quine-McCluskey Summary

1. Start with list of cubes that represent minterms where f = 1 or don't care. Generate the prime implicants by successive pairwise comparisons of the cubes.
2. Derive a cover table which indicates minterms where f = 1 that are covered by each prime implicant.
3. Include the essential prime implicants (if any) in the final cover and reduce the table by removing both these prime implicants and the covered minterms.
4. Use the concept of row and column dominance to reduce the table further. A dominated row is only removed if the cost of its prime implicant is greater than or equal to the cost of the dominating row's prime implicant.
5. Repeat steps 3 and 4 until the cover table is empty or no further reduction of the table is possible.
6. If the reduced cover table is not empty, then use the branching approach to determine the remaining prime implicants that should be included in a minimum cost cover.

# Practical Limitations of Quine-McCluskey

- Functions are seldom defined in the form of minterms, the are usually given as algebraic expressions or sets of cubes
  - List of minterms may be very large
- Many comparisons
  - Slow
- Cover table is computationally expensive