EE219B
Logic Synthesis

R. K. Brayton
University of California
Berkeley

Spring 2000

Introduction

Motivation

- Commercial success - used almost everywhere VLSI is done
- But more general –
  - Generally - discrete functions of discrete valued variables
- Body of useful and general techniques
  - can be applied to other areas

Applications

Foundation for

- Combinational and sequential logic synthesis
- Automatic test vector generation
- Timing and false paths analysis
- Formal verification
- Asynchronous synthesis
- Automata theory
- Optimal clocking schemes
- Hazard analysis
- Power estimation
- General combinatorial problems

Outline of Class

Class notes: (see class web page)

- Introduction
  - Logic functions and their representation
  - Unate recursive paradigm,
  - Two Level logic minimization (ESPRESSO)
  - Quine McCluskey method

  Midterm
- Multi-level logic synthesis
  - Introduction (Boolean networks, factored forms)
  - Division
  - Simplification
    - ☞◆●●♉⬧♓❍◻●♓➚⌧
  - Testing
  - Technology mapping
    - Advanced technology mapping

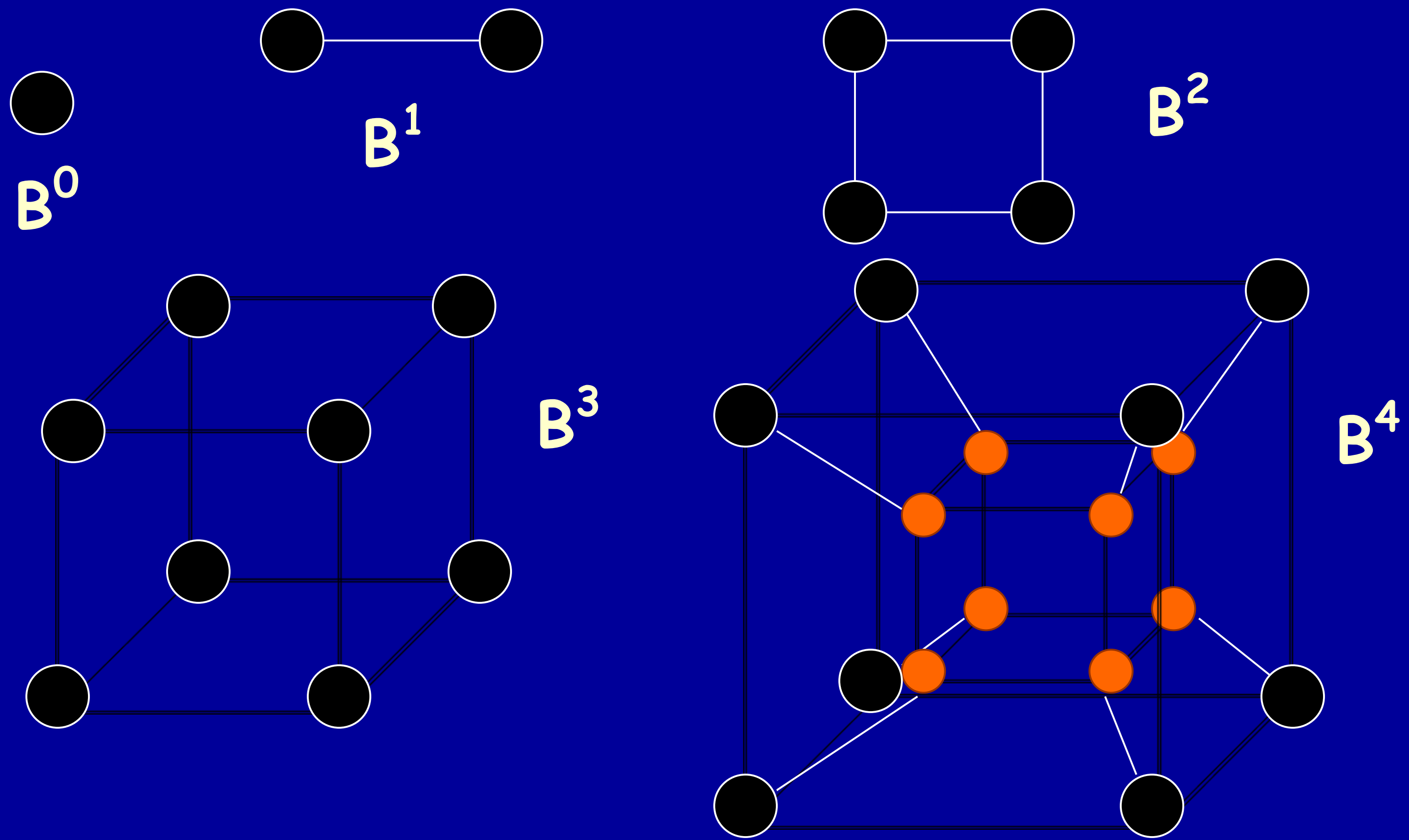
Outline of Class

- Multi-level logic synthesis (contd.)
  - Delay analysis (true and false paths)
  - Timing optimization
  - Constant delay paradigm (sizing)

  midterm
- Sequential logic synthesis
  - Introduction (FSM networks)
  - Retiming and resynthesis
  - Asynchronous synthesis
  - Node minimization?
  - ** State minimization **
- Final project presentations and report

The Boolean n-cube B^n

- $B = \{ 0,1\}$
- $B^2 = \{0,1\} \times \{0,1\} = \{00, 01, 10, 11\}$



Boolean Functions

$f(x) : B^n \longrightarrow B$

$B = \{0, 1\}, \; x = (x_1, x_2, \ldots, x_n)$

- $x_1, x_2, \ldots$ are variables
- $x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots$ are literals
- each vertex of B^n is mapped to 0 or 1
- the onset of f is $\{x|f(x)=1\} = f^1 = f^{-1}(1)$
- the offset of f is $\{x|f(x)=0\} = f^0 = f^{-1}(0)$
- if $f^1 = B^n$, f is the tautology, i.e. $f \equiv 1$
- if $f^0 = B^n$ ($f^1 = \emptyset$), f is not satisfiable
- if $f(x) = g(x)$ for all $x \in B^n$, then f and g are equivalent

We write simply f instead of f^1

Literals

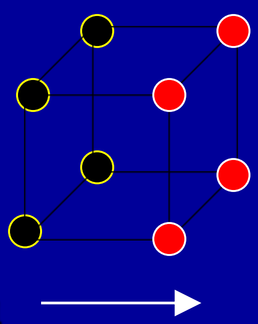
A literal is a variable or its negation

$$y, \overline{y}$$

It represents a logic function

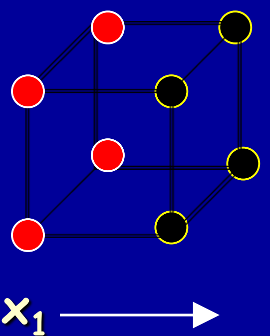
Literal x_1 represents the logic function f, where

$f = \{x \mid x_1 = 1\}$



$f = x_1$

x_1



$f = \overline{x_1}$

x_1

Literal $\overline{x_1}$ represents logic function g where

$g = \{x \mid x_1 = 0\}$

Boolean Formulas

Boolean formulas can be represented by formulas defined as catenations of

- parentheses (,)
- literals x, y, z, $\overline{x}$, $\overline{y}$, $\overline{z}$
- Boolean operators + (OR), X (AND)
- complementation, e.g. $\overline{x + y}$

Examples

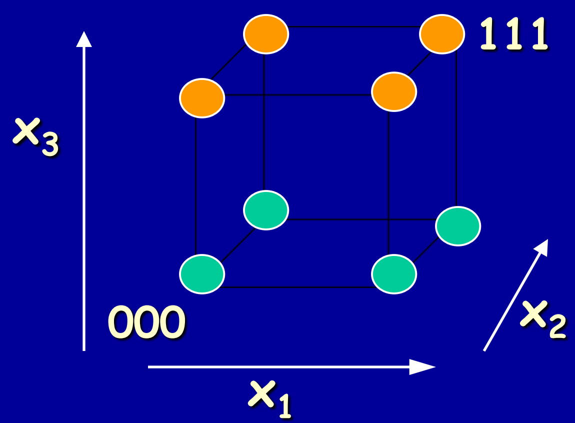
$$f = x_1 \text{ X } \overline{x_2} + \overline{x_1} \text{ X } x_2 = (x_1+x_2) \text{ X } (\overline{x_1}+\overline{x_2})$$

$$h = a + b \text{ X } c = \overline{a} \text{ X } (\overline{b} + \overline{c})$$

We usually replace X by catenation, e.g. a X b→ab

Logic functions

- There are 2^n vertices in input space B^n



000		1
001		0
010		1
011		0
100	$\Rightarrow$	1
101		0
110		1
111		0

"truth table"

- There are 2^{2^n} distinct logic functions.
  - Each subset of vertices is a distinct logic function: $f \subseteq B^n$

Logic Functions

- However, there are infinite number of logic formulas

$$
\begin{aligned}
f &= x + y \\
&= x\overline{y} + xy + \overline{x}y \\
&= x\overline{x} + x\overline{y} + y \\
&= (x + y)(x + \overline{y}) + \overline{x}y
\end{aligned}
$$

- Synthesis = Find the best formula (or "representation")

Boolean Operations - AND, OR, COMPLEMENT

$$f : B^n \rightarrow B$$

$$g : B^n \rightarrow B$$

- AND - $fg = h$ such that
  $h = \{x | f(x)=1 \text{ and } g(x)=1\}$

- OR - $f + g = h$ such that
  $h = \{x | f(x)=1 \text{ or } g(x)=1\}$

- COMPLEMENT - $\overline{f} = h$ such that
  $h = \{x | f(x) = 0\}$

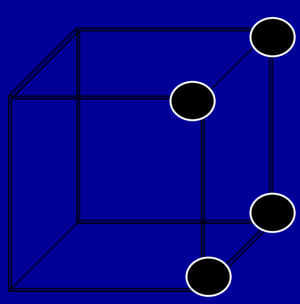
Cubes

- The AND of a set of literal functions ("conjunction" of literals) is a cube

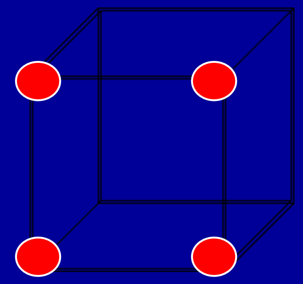
$$C = x\overline{y}$$

is a cube

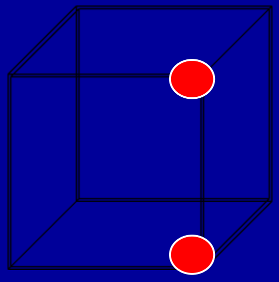
$$C = (x=1)(y=0)$$



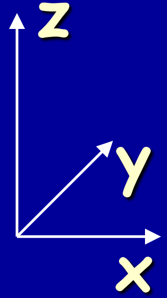
$x = 1$



$y = 0$



$x\overline{y}$



Cubes

- If $C \subseteq f$, C a cube, then C is an implicant of f.
- If $C \subseteq B^n$, and C has k literals, then $|C|$ has 2^{n-k} vertices.

Example 1 $C = x\ y \subseteq B^3$.

$k = 2$, $n = 3$.

$$C = \{100, 101\}.$$

$|C| = 2 = 2^{3-2}$.

- If $k=n$, the cube is a minterm.

Representation of Boolean functions

- The truth table of a function $f : B^n \rightarrow B$ is a tabulation of its value at each of the 2^n vertices of B^n.
- For

$$f = \overline{abc}d + \overline{ab}cd + \bar{a}b\bar{c}d + a\overline{bc}d + a\bar{b}cd + ab\bar{c}d + abc\bar{d} + abcd$$

the truth table is

	abcd	f		abcd	f
0	0000	0	8	1000	0
1	0001	1	9	1001	1
2	0010	0	10	1010	0
3	0011	1	11	1011	1
4	0100	0	12	1100	0
5	0101	1	13	1101	1
6	0110	0	14	1110	1
7	0111	0	15	1111	1

This is intractable for large n (but canonical)

Canonical means that if two functions are the same, then the canonical representations of each are isomorphic.

Sum-of-Products Representation - SOP

- A function can be represented by a sum of cubes (products):

$$f = ab + ac + bc$$

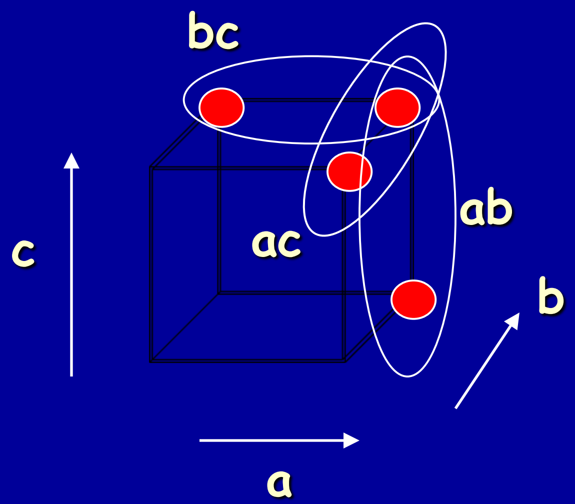
Since each cube is a product of literals, this is a "sum of products" representation

- A SOP can be thought of as a set of cubes F

$$F = \{ab, ac, bc\} = C$$

- A set of cubes that represents f is called a **cover** of f. F={ab, ac, bc} is a cover of f = ab + ac + bc.

SOP



= onset minterm

Note that each onset minterm is "covered" by at least one of the cubes, and covers no offset minterm.

- Covers (SOP's) can efficiently represent many logic functions (i.e. for many, there exist small covers).
- Two-level minimization seeks the minimum size cover (least number of cubes)

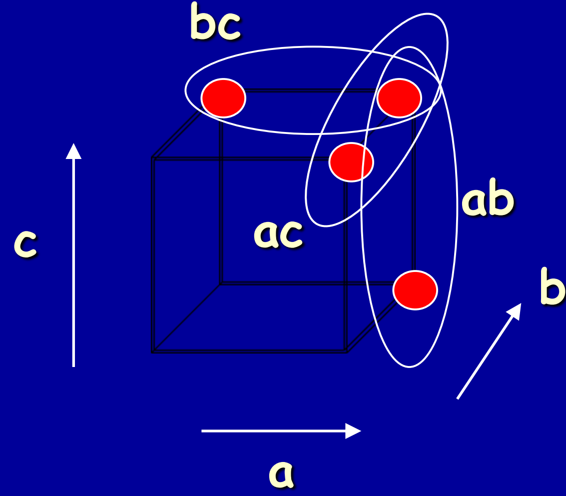
Irredundant

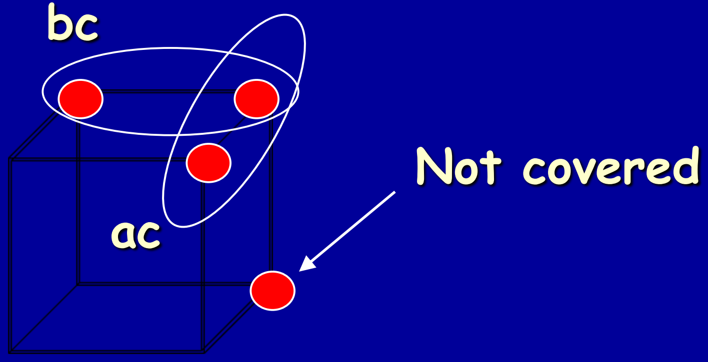
- Let $F = \{c_1, c_2, \ldots, c_k\}$ be a cover for f.

$$f = \sum_{i=1}^{k} c_i$$

A cube $c_i \in F$ is **irredundant** if $F \setminus \{c_i\} \neq f$

Example 2: $f = ab + ac + bc$





$F \setminus \{ab\} \neq f$

Prime

- A literal j of cube $c_i \in F$ ($=f$) is prime if
$$(F \setminus \{c_i\}) \cup \{c'_i\} \neq f$$
where c'_i is c_i with literal j of c_i deleted.
- A cube of F is prime if all its literals are prime.

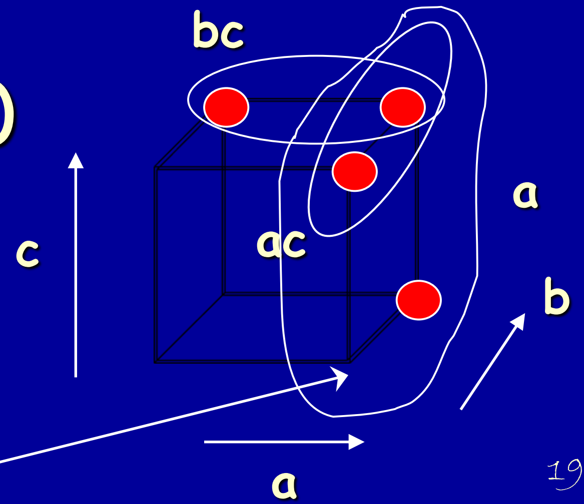
Example 3

$f = ab + ac + bc$

$c_i = ab$; $c'_i = a$ (literal b deleted)

$F \setminus \{c_i\} \cup \{c'_i\} = a + ac + bc$

$F = ac + bc + a = F \setminus \{c_i\} \cup \{c'_i\}$



Not equal to f since offset vertex is covered

Prime and Irredundant Covers

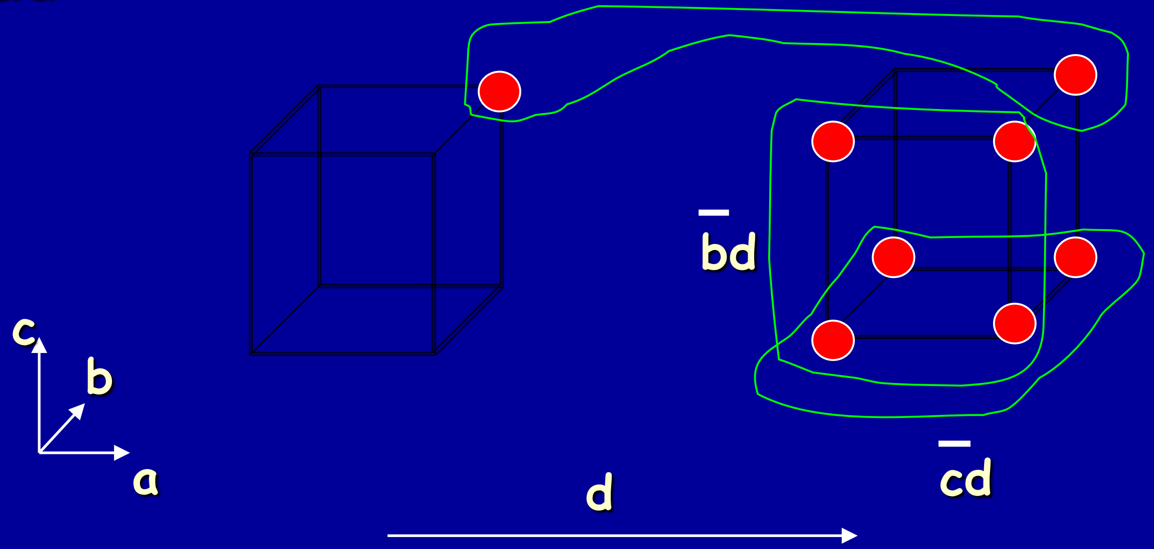
- **Definition 1** A cover is prime (irredundant) if all its cubes are prime (irredundant).
- **Definition 2** A prime of f is **essential** (essential prime) if there is a minterm (essential vertex) in that prime but in no other prime.

Prime and Irredundant Covers

Example 4

$f = abc + \bar{b}d + \bar{c}d$ is prime and irredundant.

abc is essential since $abc\bar{d} \in abc$, but not in $\bar{b}d$ or $\bar{c}d$ or ad



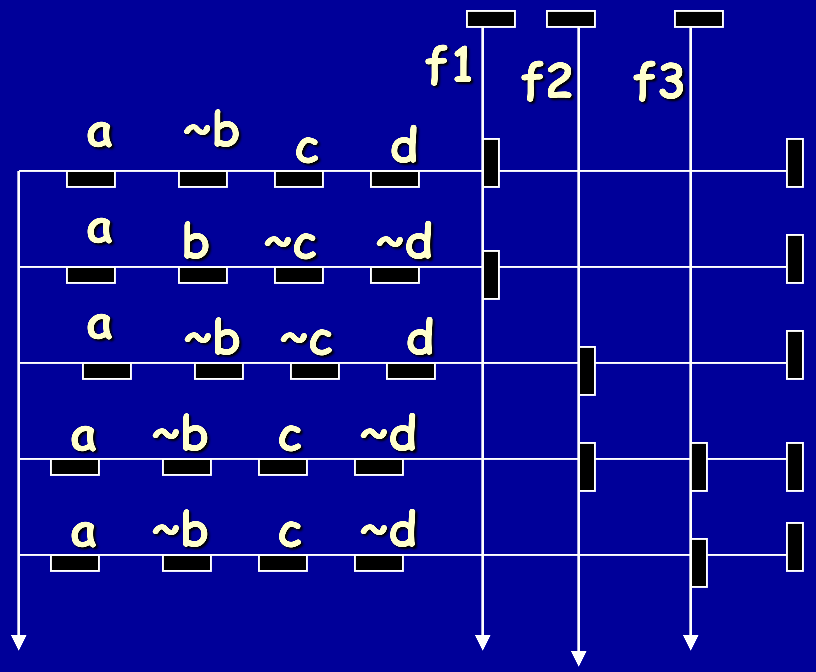
Why is abcd not an essential vertex of abc?

What is an essential vertex of abc?

What other cube is essential? What prime is not essential?

PLA's - Multiple Output Boolean Functions

- A PLA is a function $f : B^n \rightarrow B^m$ represented in SOP form:



n=4, m=3

PLA's (contd)

- Each distinct cube appears just once in the AND-plane, and can be shared by (multiple) outputs in the OR-plane, e.g., cube a~bc~d.
- Extensions from single output to multiple output minimization theory are straightforward.
- Multi-level logic can be viewed mathematically as a connection of single output functions.

Shannon (Boole) cofactors

Let $f : B^n \rightarrow B$ be a Boolean function, and $x = (x_1, x_2, \ldots, x_n)$ the variables in the support of f. The cofactor f_a of f by a literal $a=x_i$ or $a=\overline{x_i}$ is

$$f_{x_i}(x_1, x_2, \ldots, x_n) = f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)$$

$$f_{\overline{x_i}}(x_1, x_2, \ldots, x_n) = f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n)$$

Shannon (Boole) Cofactor

- The cofactor f_C of f by a cube C is just f with the fixed values indicated by the literals of C, e.g. if $C=x_i \overline{x_j}$, then $x_i =1$, and $x_j =0$.
- If $C= x_1 \overline{x_4} x_6$, f_C is just the function f restricted to the subspace where $x_1 =x_6 =1$ and $x_4 =0$.
- As a function, f_C does not depend on x_1, x_4 or x_6 (However, we still consider f_C as a function of all *n* variables, it just happens to be independent of x_1, x_4 and x_6).
- $x_1 f \neq f_{x_1}$

Example: $f= ac + \overline{a}\,\overline{c}$, $af = ac$, $f_a=c$

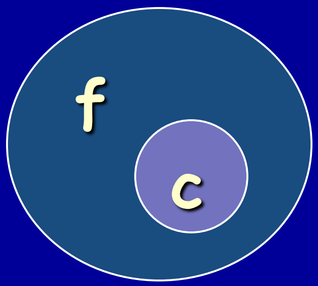
Fundamental Theorem

Theorem 1 Let c be a cube and f a function. Then $c \subseteq f \Leftrightarrow f_c \equiv 1$.

Proof. We use the fact that $xf_x = xf$, and f_x is independent of x.

If: Suppose $f_c \equiv 1$. Then $cf=f_cc=c$. Thus,

$c \subseteq f$.



Proof (contd)

Only if. Assume $c \subseteq f$

Then $c \subseteq cf = cf_c$. But f_c is independent of literals $l \in c$. If $f_c \neq 1$, then $\exists\, m \in B^n$, $f_c(m)=0$.

Let $m_i' = m_i$, if $x_i \notin c$ and $\overline{x}_i \notin c$.

or if $m_i=0$, $\overline{x}_i \in c$

or $m_i=1$, $x_i \in c$.

$m_i' = \overline{m}_i$ otherwise.

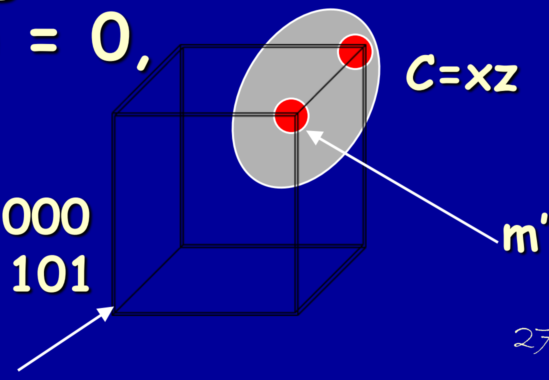
i.e. we make the literals of m' agree with c, i.e. $m' \in c$. But then $f_c(m') = f_c(m) = 0$,

Hence, $c(m')=1$

and $f_c(m')\, c(m') = 0$,

contradicting $c \subseteq cf_c$.

$m = 000$
$m' = 101$



End of Lecture 1

Cofactor of Covers

Definition 3 The **cofactor of a cover** F is the sum of the cofactors of each of the cubes of F.

Note: If $F=\{c_1, c_2, \ldots, c_k\}$ is a cover of f, then $F_c=\{(c_1)_c, (c_2)_c, \ldots, (c_k)_c\}$ is a cover of f_c.

Suppose F(x) is a cover of f(x), i.e.

$$F(x) = \sum_i c_i = \sum_i \prod_j \ell^i_j = \{c_i\}$$

Then for $1 \le j \le n$,

$$F(x)_{x_j} = \sum_i (c_i)_{x_j}$$

is a cover of $f_{x_j}(x)$

Definition 4 The cofactor C_{x_j} of a cube C with respect to a literal x_j is

- C if x_j and $\overline{x_j}$ do not appear in C
- $C \backslash \{x_j\}$ if x_j appears positively in C, i.e. $x_j \in C$
- $\varnothing$ if x_j appears negatively in C, i.e. $\overline{x_j} \in C$

Example 5

If $C = x_1 \overline{x_4} x_6$,

$C_{x_2} = C$ (x_2 and $\overline{x_2}$ do not appear in C)

$C_{x_1} = \overline{x_4} x_6$ (x_1 appears positively in C)

$C_{x_4} = \varnothing$ (x_4 appears negatively in C)

Example 6

$F = abc + \bar{b}d + \bar{c}d$

$F_b = ac + \bar{c}d$

(Just drop b everywhere and throw away cubes containing literal $\bar{b}$)

Shannon Expansion

$$f : B^n \rightarrow B$$

Theorem 2 $$f = x_i f_{x_i} + \overline{x}_i f_{\overline{x}_i}$$

Theorem 3 F is a cover of f. Then

$$\tilde{F} \equiv x_i F_{x_i} + \overline{x}_i F_{\overline{x}_i}$$

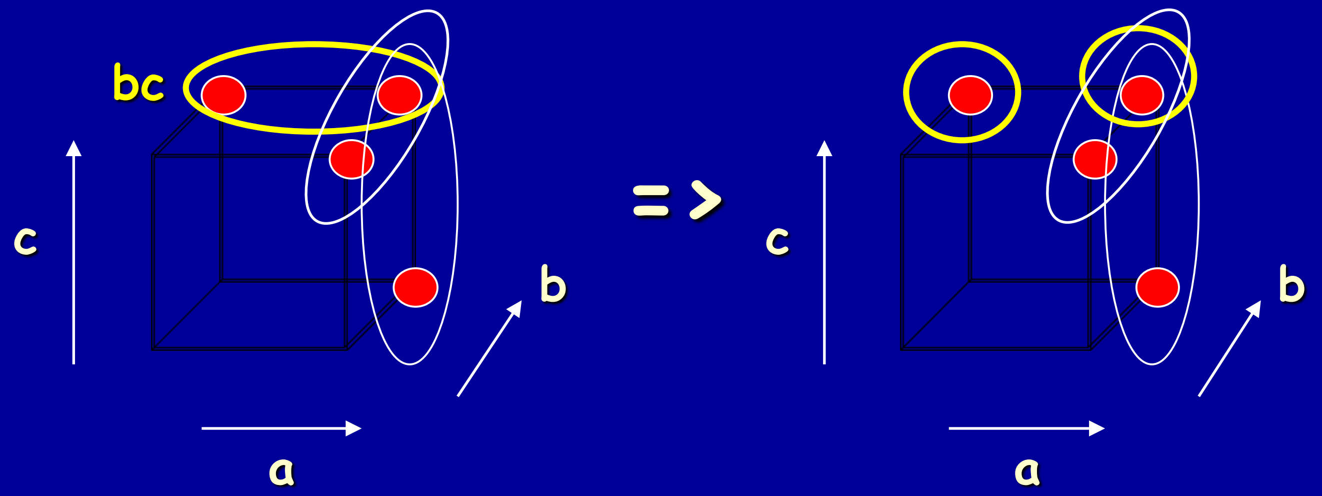
We say that f (F) is expanded about x_i. x_i is called the splitting variable.

Example 7

$$F = ab + ac + bc$$

$$\tilde{F} = aF_a + \bar{a}F_{\bar{a}} = a(b + c + bc) + \bar{a}(bc)$$

$$= ab + ac + abc + \bar{a}bc$$



Cube bc got split into two cubes

Cover Matrix

We sometimes use matrix notation to represent a cover:

Example 8 $F = ac + \bar{c}d =$

$$\begin{array}{r} \\ ac \rightarrow \\ \bar{c}d \rightarrow \end{array} \begin{array}{c} a\ b\ c\ d \\ \boxed{\begin{array}{cccc} 1 & 2 & 1 & 2 \\ 2 & 2 & 0 & 1 \end{array}} \end{array} \quad \text{or} \quad \begin{array}{c} a\ b\ c\ d \\ \boxed{\begin{array}{cccc} 1 & - & 1 & - \\ - & - & 0 & 1 \end{array}} \end{array}$$

Each row represents a cube. 1 means that the positive literal appears in the cube, 0 the negative. The 2 (or -) here represents that the variable does **not appear** in the cube. It also represents both 0 and 1 values.

Example 9

$a\overline{c}$ = 1 2 0 2 = {1, {0,1}, 0, {0,1}}

= {1000, 1100, 1001, 1101}

2 is sometimes called "input don't care", but this is confusing so we won't use the term.

Incompletely Specified Functions

$F = (f, d, r) : B^n \rightarrow \{0, 1, *\}$

where * represents "don't care". (Sometimes we use 2 in place of *)

- f = onset function - $f(x)=1 \leftrightarrow F(x)=1$
- r = offset function - $r(x)=1 \leftrightarrow F(x)=0$
- d = don't care function - $d(x)=1 \leftrightarrow F(x)=*$

(f,d,r) forms a partition of B^n. i.e.

- $f + d + r = B^n$
- $fd = fr = dr = \emptyset$ (pairwise disjoint)

A completely specified function g is a **cover** for $F=(f,d,r)$ if

$$f \subseteq g \subseteq f+d$$

(Thus $gr = \emptyset$). Thus, if $x \in d$ (i.e. $d(x)=1$), then $g(x)$ can be 0 or 1, but if $x \in f$, then $g(x)=1$ and if $x \in r$, then $g(x)=0$.

(We "don't care" which value g has at $x \in d$)

Primes of Incompletely Specified Functions

Definition 5 A cube c is prime of $F=(f,d,r)$ if $c \subseteq f+d$ (an implicant of $f+d$), and no other implicant (of $f+d$) contains c, i.e.

$$\forall \tilde{c}, \tilde{c} \subseteq f + d, c \not\subset \tilde{c}$$

(i.e. it is simply a prime of $f+d$)

Definition 6 Cube c_j of cover $F=\{c_i\}$ is redundant if $f \subseteq F \setminus \{c_j\}$. Otherwise it is irredundant.

Note that $c \subseteq f+d \leftrightarrow cr = \varnothing$

Example:Logic Minimization

Consider $F(a,b,c)=(f,d,r)$, where $f=\{\bar{a}b\bar{c},\ a\bar{b}c,\ abc\}$ and $d=\{a\bar{b}\bar{c},\ ab\bar{c}\}$, and the sequence of covers illustrated below:

$F^1 = \bar{a}b\bar{c} + a\bar{b}c + abc$

Expand $abc \to a$

$F^2 = a + \bar{a}b\bar{c} + a\bar{b}c$

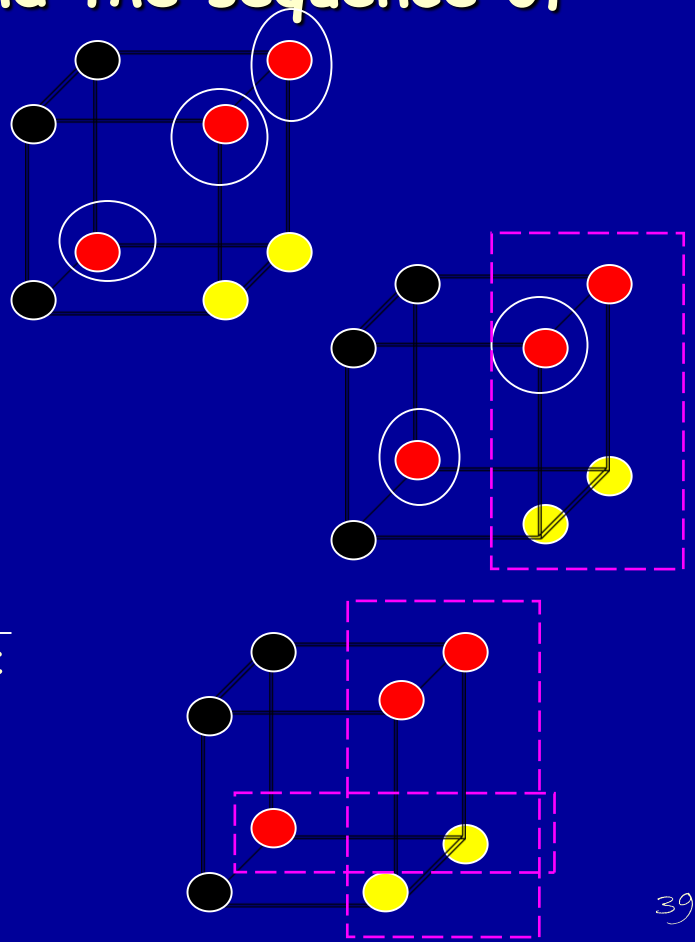
$a\bar{b}c$ is redundant

a is prime

$F^3 = a + \bar{a}b\bar{c}$

Expand $\bar{a}b\bar{c} \to b\bar{c}$

$F^4 = a + b\bar{c}$



- on
- off
- Don't care

Checking for Prime and Irredundant

Let $G=\{c_i\}$ be a cover of $F=(f,d,r)$. Let D be a cover for d.

- $c_i \subseteq G$ is **redundant** iff

$$c_i \subseteq (G \setminus \{c_i\}) \cup D \equiv G^i \qquad (1)$$

(Since $c_i \subseteq G^i$ and $f \subseteq G \subseteq f+d$ then $c_i \subseteq c_i f + c_i d$ and $c_i f \subseteq G \setminus \{c_i\}$. Thus $f \subseteq G \setminus \{c_i\}$.)

- A literal $l \in c_i$ is **prime** if $(c_i \setminus \{ l \})$ ($= (c_i)_l$) is not an implicant of F.
- A cube c_i is a prime of F iff all literals $l \in c_i$ are prime.

Literal $l \in c_i$ is not prime $\Leftrightarrow (c_i)_l \subseteq f+d$ $\qquad$ (2)

Note: Both tests (1) and (2) can be checked by tautology:

1) $(G^i)_{c_i} \equiv 1$ (implies c_i redundant)
2) $(F \cup D)_{(c_i)_l} \equiv 1$ (implies l not prime)

Example

$F=\{\bar{a}b\bar{c},\ a\bar{b}c,\ abc\}$

$D=\{a\bar{c}\}$

$R=\{\bar{a}\bar{b}, \bar{a}c\}$

Expand $abc \rightarrow ab = (c_i)_l = (abc)_c$

Check $ab \subseteq f+d \ \Leftrightarrow (f+d)_{ab} = 1$

- $(f+d)_{ab} = c + \bar{c} \equiv 1$ OK

Expand $ab \rightarrow a$. Check $a \subseteq f+d$

- $(f+d)_a = \bar{b}c + bc + \bar{c} \equiv 1$ OK

$F = a + a\bar{b}c + \bar{a}b\bar{c}$

Check if $a\bar{b}c$ is redundant

- $a\bar{b}c \subseteq a + \bar{a}b\bar{c} + a\bar{c} = F \backslash \{a\bar{b}c\} \cup D$
- $(a + \bar{a}b\bar{c} + a\bar{c})_{a\bar{b}c} = 1$ OK

Cover is now $F = a + \bar{a}b\bar{c}$ and $D = a\bar{c}$

Check if $\bar{a}b\bar{c}$ is redundant

$\bar{a}b\bar{c} \subseteq a + a\bar{c} = F \setminus \{\bar{a}b\bar{c}\} \cup D$

- $(a + a\bar{c})_{\bar{a}b\bar{c}} = \emptyset$ NOT REDUNDANT