ECE 3060
VLSI and Advanced Digital Design

Lecture 10

Two Level Logic Minimization

Motivation

- We will study modern techniques for manipulating and minimizing boolean functions
- Issue: Tractibility of minimization problem for large number of variables
  - Exact methods
  - Heuristic methods
- Issue: Representation of boolean expressions in a form conducive to boolean operations
  - Implicant tables
  - Binary decision diagrams
- Issue: Manipulation of realistic multilevel networks
  - Graph representations
  - multilevel minimization
  - Technology mapping

Definitions

- **Binary space** $B = \{0, 1\}$
- **Operations OR(+), AND(.), NOT**
- **Single output:** $f: B^n \rightarrow B$
- **Incompletely specified single output function:**

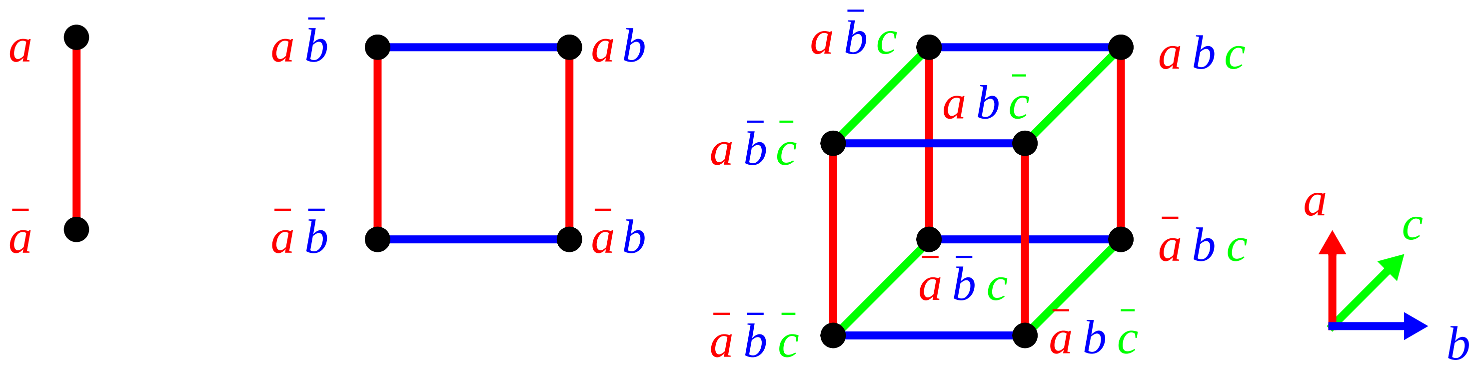
$$f: B^n \rightarrow \{0, 1, *\}$$

- **Multiple output:** $f: B^n \rightarrow B^m$
- **Incompletely specified multiple output function:**

$$f: B^n \rightarrow \{0, 1, *\}^m$$

Cube Representation

- B^n **can be represented by a binary n-cube, i.e. an n-dimensional binary hypercube**



- **As usual, literals may be replaced with binary values,**

  i.e. $a\bar{b}c \equiv 010$

- **Adjacent minterms (vertices) differ in only one variable similar to K-map**

Definitions

- **Boolean *variable*:** $a \in B$
- **Boolean *literal*:** a **or** $\bar{a}$
- ***Product* or *cube*: product of literals**
- ***Implicant*: product term implying a value of a function (usually TRUE)**
  - **binary hypercube in the boolean space**
- ***Minterm*: product using all input variables implying a value of a function (usually TRUE)**
  - **vertex in the boolean space**

Tabular Representations

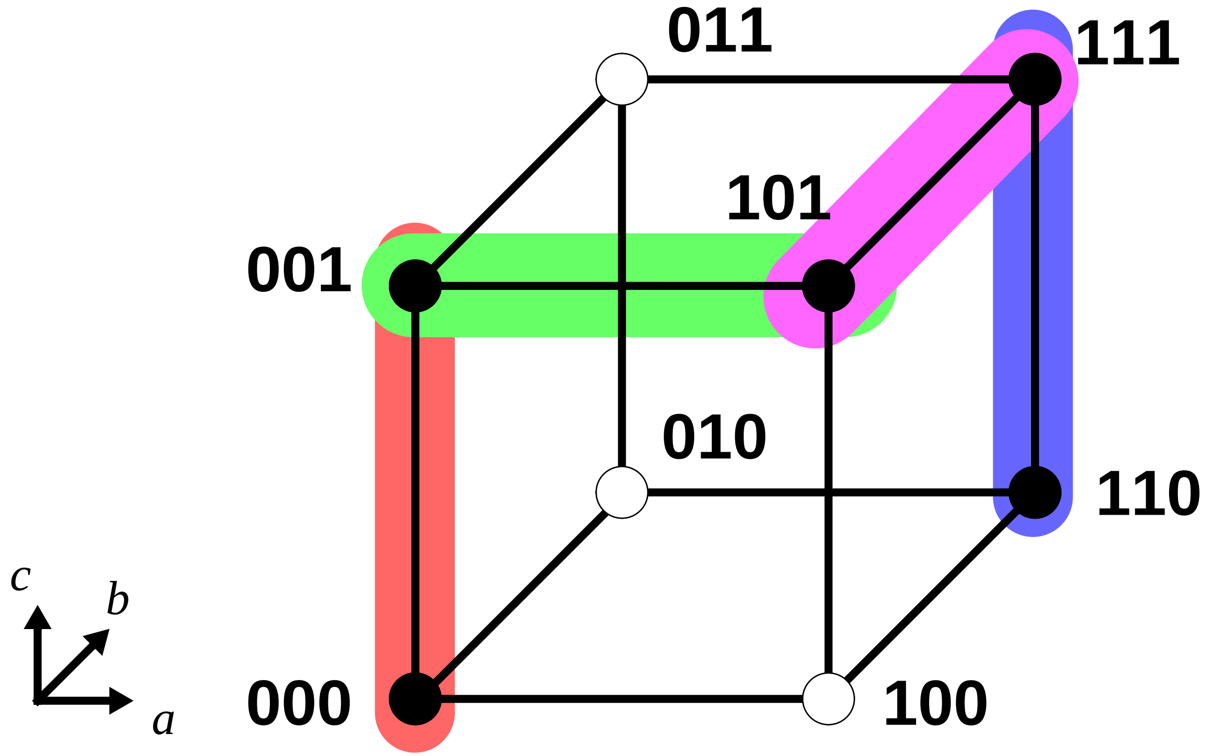
- ***Truth table***
  - **list of all minterms of a function**
- ***Implicant table*** **or** ***cover***
  - **list of all implicants of a function sufficient to define the function**
- **Comment:**
  - **Implicant tables are smaller in size**
- **Example Cover**

abc	xy
001	10
*11	11
101	01
11*	11

$$x = ab + \bar{a}c$$

$$y = ab + bc + ac$$

Cube Representation



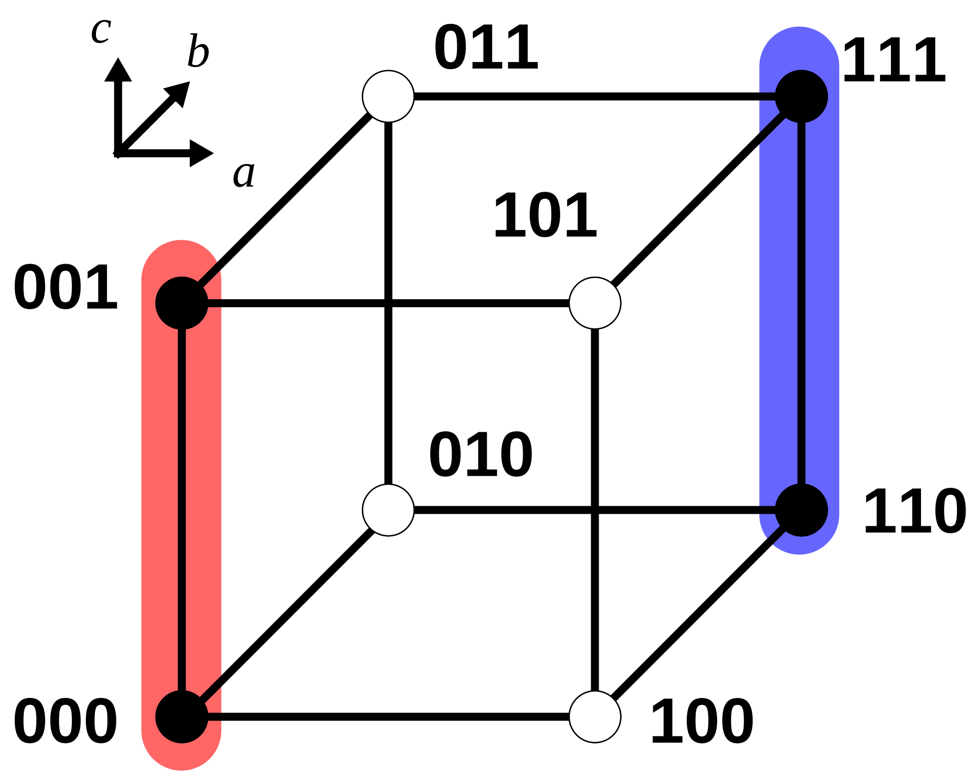
- $F = \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + a\bar{b}c + abc + ab\bar{c}$
- $F = \bar{a}\bar{b} + \bar{b}c + ac + ab$

Prime Definitions

- **Prime implicant**
  - **implicant not contained by any other implicant**
- **Prime cover**
  - **cover of prime implicants**
- **Essential Prime Implicant (EPI):**
  - **there is at least one minterm covered by EPI and not covered by any other prime implicant**

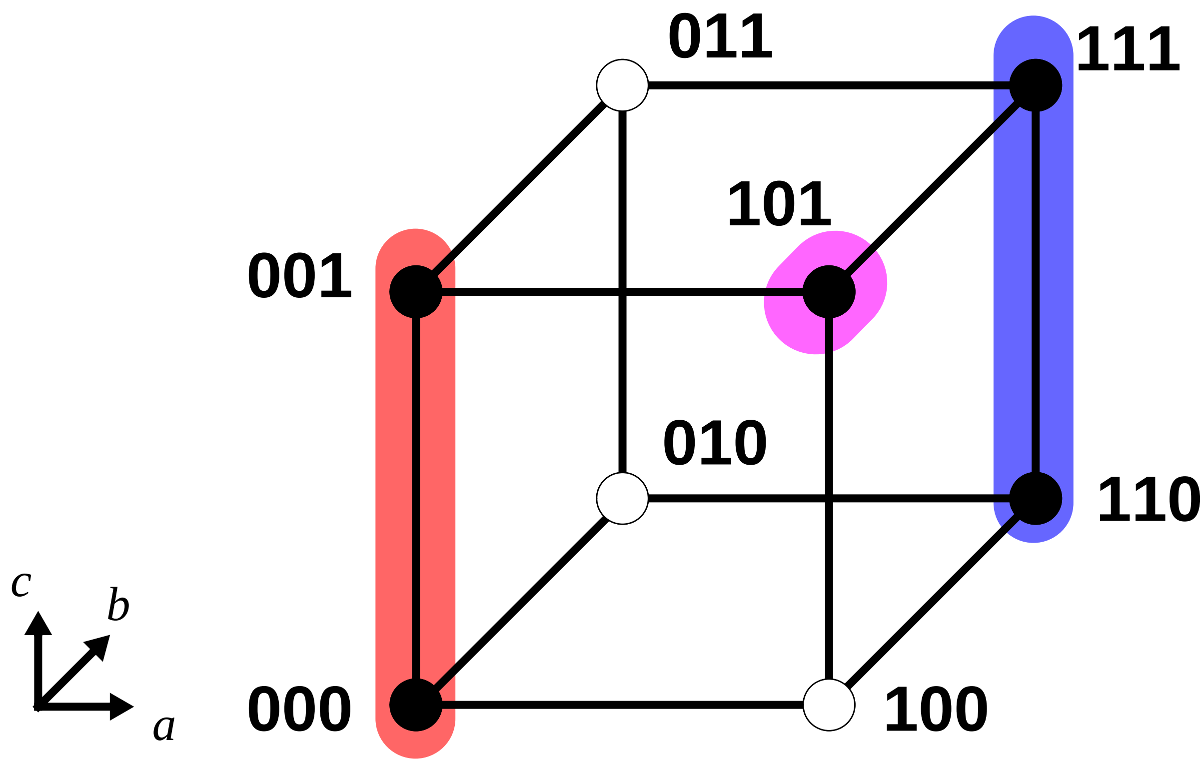
Two level logic optimization

- **Assumptions:**
  - **primary goal is to reduce the number of implicants**
  - **all implicants have the same cost**
  - **secondary goal is to reduce the number of literals**
- **Minimum cover**
  - **cover of the function with the minimum number of implicants**
  - **global optimum**
- $f\ =$



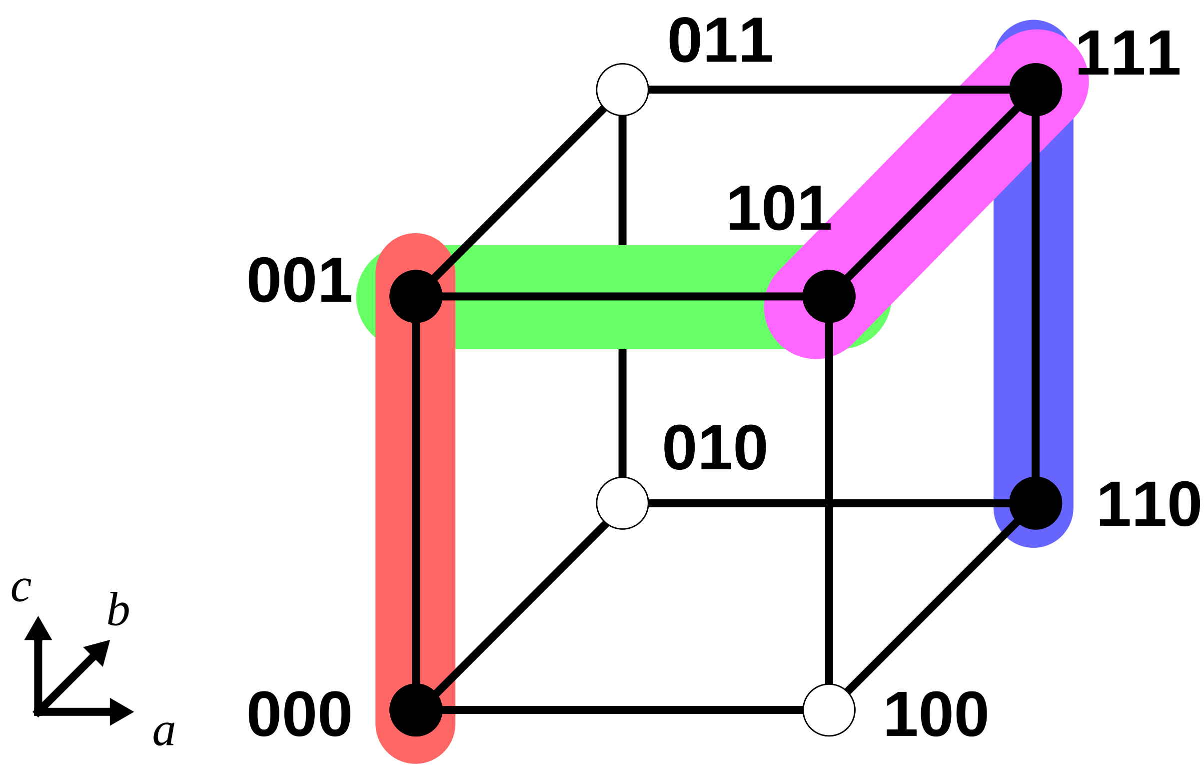
Minimal or Irredundant Cover

- **Cover of the function that is not a proper superset of another cover**
  - **no implicant can be dropped**
  - **local optimum**



Minimal Cover with respect to single-implicant containment

- no implicant is contained by any other implicant
- weak local optimum



Logic Minimization

- *Exact* methods:
  - compute minimum cover
  - often intractable for large functions
  - based on Quine-McCluskey method
- *Heuristic* methods:
  - Compute minimal cover (possibly minimum)
- There are a large variety of methods and programs
  - academic: MINI, PRESTO, ESPRESSO (UCBerkeley)
  - industry: Synopsys, Cadence, Mentor Graphics, Zuken

Exact Logic Minimization

- ***Quine's*** **theorem:**
  - **There is a minimum cover that is prime**
  - **Consequently, the search for minimum cover can be restricted to prime implicants**
- ***Quine-McCluskey*** **method:**
  1. compute prime implicants
  2. determine minimum cover via branching
- ***Petrick's*** **method**
  1. compute prime implicants
  2. determine minimum cover via covering clause

Computing Prime Implicants

- **The *Hamming weight* of a minterm is the number of ones in that minterm.**
- **Start with list of minterms sorted by Hamming weight.**

1. Combine all possible implicants (minterms) using $\alpha y + \alpha \bar{y} = \alpha$. Note that this algebraic reduction specifies two implicants with Hamming weights that differ by one.
2. Group resulting implicants by Hamming weight.
3. Repeat 1. and 2. on the resulting implicants until no further factoring is possible (i.e. all implicants are prime)

- **Example:**

$$f = \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}\bar{c}d + \bar{a}\bar{b}c\bar{d} + \bar{a}b\bar{c}d + \bar{a}bc\bar{d} + \bar{a}bcd$$
$$+ a\bar{b}\bar{c}\bar{d} + a\bar{b}\bar{c}d + a\bar{b}c\bar{d} + abc\bar{d}$$

Prime Implicant Table

- **Rows: minterms**
- **Columns: prime implicants**
- **Exponential size: for a function** $f: B^n \to B$
  - 2^n **minterms**
  - **up to** $3^n / n$ **prime implicants**

Example

- **Function:** $f = \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + a\bar{b}c + abc + ab\bar{c}$

- **Primes:**

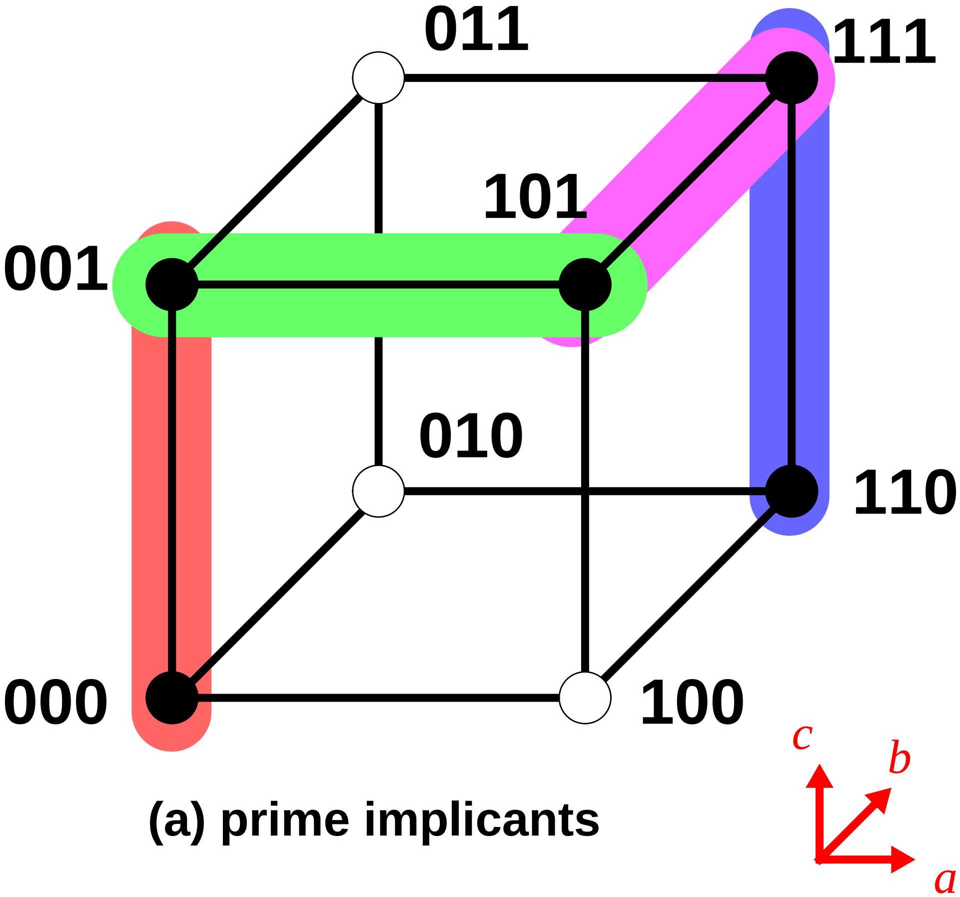
Label	PIs
α	00*
β	*01
γ	1*1
δ	11*

- **Implicant Table:**

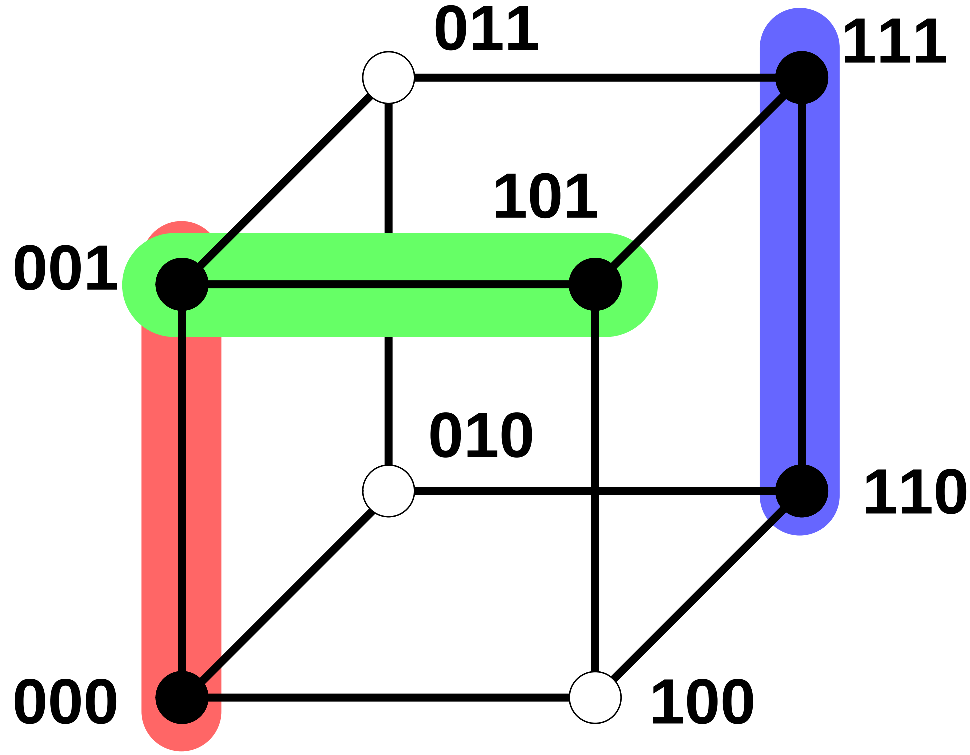
Minterms	Primes			
	α	β	γ	δ
$\bar{a}\bar{b}\bar{c}$	1	0	0	0
$\bar{a}\bar{b}c$	1	1	0	0
$a\bar{b}c$	0	1	1	0
abc	0	0	1	1
$ab\bar{c}$	0	0	0	1

- **Choose cover by selecting a set of implicants which *cover* all minterms.**

Cube Representation



(a) prime implicants



(b) minimum cover

Petrick's Method

- **Determine minimum cover via covering clause:**
    1. Write covering clauses in *pos* form
    2. Multiply out *pos* form into *sop* form (and simpify).
    3. Select cube of minimum size
- **Covering clause describes necessary and sufficient conditions to cover function**
- **Note: multiplying out clauses is exponential.**
- **Example:**
    - ***pos* clauses:** $(\alpha)(\alpha + \beta)(\beta + \gamma)(\gamma + \delta)(\delta)$ **(Each term covers a minterm)**
    - ***sop* clauses:** $\alpha\beta\delta + \alpha\gamma\delta$
    - **Covers:** $\{\alpha, \beta, \delta\}$ **or** $\{\alpha, \gamma, \delta\}$

Exact Two-level Logic Minimization

- Matrix representation
- Covering problem
- Reduction strategies
- Branch and bound covering algorithm

Matrix representation

- **View implicant table of some function f as Boolean matrix: A**
  - $(a_{ij} = 1) \Rightarrow$ **the ith minterm is covered by the jth prime implicant**
- **The (Boolean) selection vector x selects which prime implicants will be in the cover.**
- **To cover f, find an x which satisfies**

$$y_i \geq 1 \forall i \qquad Ax = y$$

  - **i.e. select enough columns to cover all rows**
- **To find a minimum cover, minimize cardinality of x, i.e. the number of nonzero entries of x.**

Example

- **The magnitude of y_i indicates the number of prime implicants which cover the ith minterm.**

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Branch and Bound Algorithm

- **Exact algorithm, but not polynomial time.**
- **First step:**
- **Remove Essential Prime Implicants (EPIs) which are columns incident to one (or more) row(s) with a single 1 in them.**
- **Modify A by removing the column and incident rows**
- **Example: rows 1 and 5 from previous matrix**

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = A \text{ **becomes** } \begin{bmatrix} 1 & 1 \end{bmatrix}$$

Reduction Strategies

- **Column (implicant) dominance:**
  - **a column** i ***dominates*** **column** j **iff, for all rows** k**,** $a_{ki} \geq a_{kj}$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

  - **In this example, which columns dominate?**
- **any** ***dominated*** **column** j **may be removed, because the implicant corresponding to column** j **is not prime**

Reduction Strategies

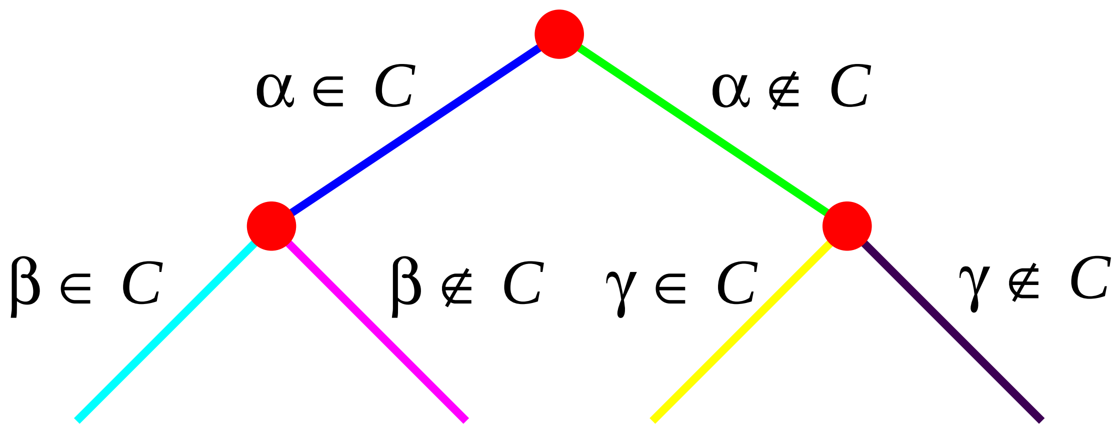
- **Row (minterm) dominance**
  - **a row k dominates row l iff, for all columns i** $a_{ki} \geq a_{li}$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

  - **Which rows dominate?**
- **a row k, which *dominates* another row l, may be removed because whichever implicant is eventually chosen to cover l will also cover k**

Branching Algorithm

- Remove essential primes from consideration.
- Perform a depth-first search of remaining covers.
- Bounding algorithm used to prune search tree.



Branch and Bound Algorithm

EXACT_COVER(A, x, b) { /* b is current best estimate */

  Reduce matrix A and update corresponding x;

  Calculate *current_estimate* for this branch; /* we don't cover this */

  if (*current_estimate* >= $|b|$) return (b);

  if (A has no rows) return (x);

  Select a branching column c;

  $x_c = 1$; /* this changes element c in x */

  $\tilde{A} = A$ after deleting column c and rows incident to c;

  $\tilde{x}$ = EXACT_COVER($\tilde{A}, x, b$);

  if $(|\tilde{x}| < |b|)$ $b = \tilde{\tilde{x}}$;

  $x_c = 0$;

  $\tilde{A} = A$ after deleting column c;

  $\tilde{x}$ = EXACT_COVER($\tilde{A}, x, b$);

  if $(|\tilde{x}| < |b|)$ $b = \tilde{x}$;

  return (b);

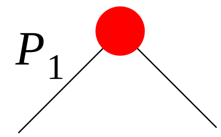
}

Example

- **Consider** $A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$**,** $x = \begin{bmatrix} x1 \\ x2 \\ x3 \\ x4 \\ x5 \end{bmatrix}$**,** $b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

  There are no essential primes, and no row or column dominance.
- **Denote the implicants** P_j **and the minterms as** μ_i
- **Choose** P_1 **(i.e.** $c = 1$**)**

- **Now** $\tilde{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$, $\tilde{x} = \begin{bmatrix} 1 \\ x2 \\ x3 \\ x4 \\ x5 \end{bmatrix}$

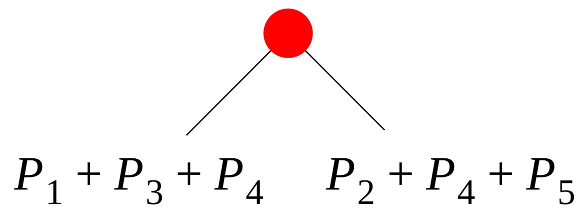


- **Columns 2 and 5 are dominated; after removing columns 2 and 5, row 3 is dominating so I_3 and I_4 covering μ_2 and μ_3 are now essential so we get**

$$\tilde{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \ \tilde{x} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$ **and since** $|\tilde{x}| < |b|$, $b = x$

- **Now we consider the solution without** P_1

- **Now** $\tilde{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$**,** $\tilde{x} = \begin{bmatrix} 0 \\ 1 \\ x3 \\ x4 \\ x5 \end{bmatrix}$



- **Now column 2 is essential, leaving column 3 dominated, and row 4 is dominating, leaving columns 4 and 5 as essentials.**

Reading

- **Read the draft chapters of Professor Mooney's book**
- **If you are curious, here are some advanced texts:**
  - **<u>Synthesis and Optimization of Digital Circuits</u>, Giovanni De Micheli, McGraw-Hill, 1994.**
  - **<u>Logic Synthesis and Verification Algorithms</u>, Gary Hachtel and Fabio Somenzi, Kluwer Academic Publishers, 1994.**
- **Obviously, if you are interested in VLSI beyond what this course presents, you may take ECE 4130.**
- **Also, if you are interested in synthesis beyond what this course presents, you may want to take ECE 6132 Computer-Aided VLSI System Design, a new graduate course which undergrads can take with the permission of Professor Mooney.**