

Logic Synthesis and Verification

Jie-Hong Roland Jiang
江介宏

Department of Electrical Engineering
National Taiwan University



Fall 2013

1

Two-Level Logic Minimization (1/2)

Reading:

Logic Synthesis in a Nutshell

Section 3 (§3.1-§3.2)

most of the following slides are by
courtesy of Andreas Kuehlmann

2

Quine-McCluskey Procedure

- Given G and D (covers for $\mathfrak{S} = (f, d, r)$ and d , respectively), find a minimum cover G^* of primes where:
 $f \subseteq G^* \subseteq f+d$ (G^* is a prime cover of $\mathfrak{S}$)
- Q-M Procedure:
 1. Generate all primes of $\mathfrak{S}$, $\{P_j\}$ (i.e. primes of $(f+d) = G+D$)
 2. Generate all minterms $\{m_i\}$ of $f = G \wedge \neg D$
 3. Build Boolean matrix B where

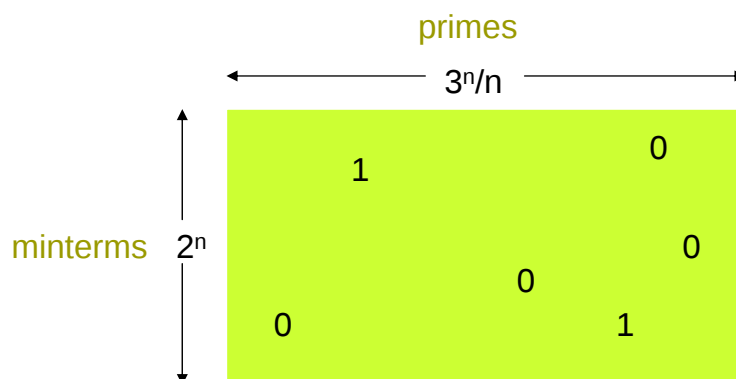
$$B_{ij} = 1 \text{ if } m_i \in P_j$$

$$= 0 \text{ otherwise}$$
 4. Solve the minimum column covering problem for B (unate covering problem)

3

Complexity

- $\sim 2^n$ minterms; $\sim 3^n/n$ primes



- There are $O(2^n)$ rows and $\Omega(3^n/n)$ columns. Moreover, minimum covering problem is NP-complete. (Hence the complexity can probably be double exponential in size of input, i.e. difficulty is $O(2^{3^n})$)

4

Two-Level Logic Minimization

Example

Karnaugh map

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

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

$$F = \bar{x} \bar{y} z w + \bar{x} y \bar{z} w + x \bar{y} z w + \bar{x} y z w \quad (\text{cover of } \mathfrak{F})$$

$$D = \bar{y} z + xyw + \bar{x} \bar{y} z w + x \bar{y} w + \bar{x} y \bar{z} w \quad (\text{cover of } d)$$

Primes: $\bar{y} + w + \bar{x} \bar{z}$

Covering Table

Solution: $\{1, 2\} \Rightarrow \bar{y} + w$ is a minimum prime cover (also $w + \bar{x} \bar{z}$)

5

Covering Table

	$\bar{y}$	w	$\bar{x} \bar{z}$	Primes of f+d
$\bar{x} \bar{y} \bar{z} \bar{w}$	1	0	1	
$\bar{x} y \bar{z} w$	0	1	1	
$x \bar{y} \bar{z} w$	1	1	0	
$\bar{x} y z w$	0	1	0	Row singleton (essential minterm)

↑
Essential prime

Definition. An **essential prime** is a prime that covers an onset minterm of f not covered by any other primes.

6

Covering Table

Row Equality

□ Row equality:

- In practice, many rows in a covering table are identical. That is, there exist minterms that are contained in the same set of primes.

■ Example

m_1	0101101
m_2	0101101

7

Covering Table

Row and Column Dominance

□ Row dominance:

- A row i_1 whose set of primes is contained in the set of primes of row i_2 is said to **dominate** i_2 .

■ Example

i_1	011010
i_2	011110

- i_1 dominates i_2
- Can remove row i_2 because have to choose a prime to cover i_1 , and any such prime also covers i_2 . So i_2 is automatically covered.

8

Covering Table

Row and Column Dominance

□ Column dominance:

- A column j_1 whose rows are a superset of another column j_2 is said to **dominate** j_2 .

■ Example

j_1	j_2
1	0
0	0
1	1
0	0
1	1

- j_1 dominates j_2
- We can remove column j_2 since j_1 covers all those rows and more. We would never choose j_2 in a minimum cover since it can always be replaced by j_1 .

9

Covering Table

Table Reduction

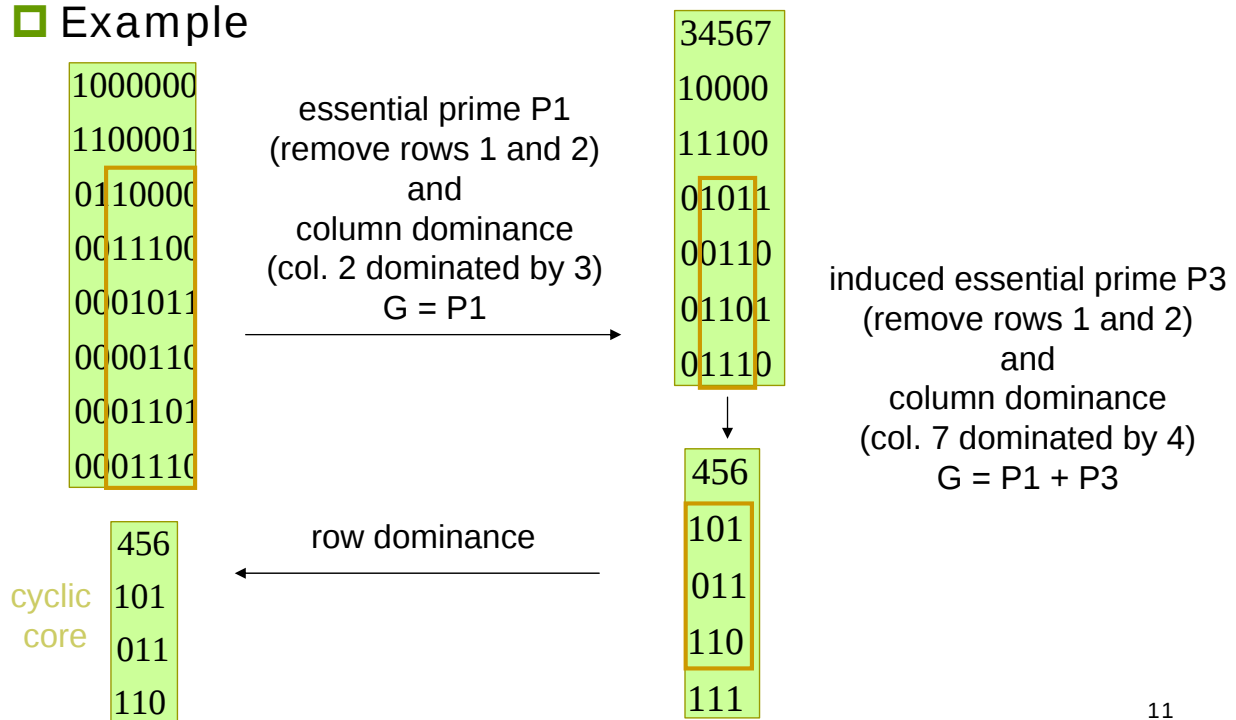
1. Remove all rows covered by essential primes (columns in row singletons). Put these primes in the cover G .
2. Group identical rows together and remove dominated rows.
3. Remove dominated columns. For equal columns, keep one prime to represent them.
4. Newly formed row singletons define **induced essential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

- The resulting reduced covering table is called the **cyclic core**. This has to be solved (**unate covering problem**). A minimum solution is added to G . The resulting G is a minimum cover.

10

Covering Table Table Reduction

Example



11

Solving Cyclic Core

- Best known method (for unate covering) is **branch and bound** with some clever bounding heuristics

Independent Set Heuristic:

- Find a maximum set I of "independent" rows. Two rows B_{i_1}, B_{i_2} are independent if **not** $\exists j$ such that $B_{i_1j} = B_{i_2j} = 1$. (They have **no column in common.**)

Example

A covering matrix B rearranged with independent sets first

$B =$

1 1 1	
1 1 1 1	0
1 1	
A	C

Independent set $\mathcal{I}$ of rows

12

Solving Cyclic Core

□ Lemma:

$$|\text{Solution of Covering}| \geq |\mathcal{J}|$$

m_1 must be covered by one of the three columns

m_1	1 1 1	
m_2	1 1 1 1	0
m_3	1 1	
	A	C

13

Solving Cyclic Core

□ Heuristic algorithm:

- Let $\mathcal{J} = \{l_1, l_2, \dots, l_k\}$ be the independent set of rows
- 1. choose $j \in l_i$ such that column j covers the most rows of A. Put P_j in G
- 2. eliminate all rows covered by column j
- 3. $\mathcal{J} \leftarrow \mathcal{J} \setminus \{l_i\}$
- 4. go to 1 if $|\mathcal{J}| > 0$
- 5. If B is empty, then done (in this case achieve minimum solution because of the lower bound of previous lemma attained - **IMPORTANT**)
- 6. If B is not empty, choose an independent set of B and go to 1

1 1 1	
1 1 1 1	0
1 1	
A	C

14

Prime Generation for Single-Output Function

Tabular method

(based on *consensus* operation, or ∇):

- Start with minterm canonical form of F
- Group *pairs* of adjacent minterms into cubes
- Repeat merging cubes until no more merging possible; mark (✓) + remove all covered cubes.
- Result: set of *primes* of f .

Example

$$F = x' y' + w x y + x' y z' + w y' z$$

$$F = x' y' + w x y + x' y z' + w y' z$$

$w' x' y' z'$ ✓	$w' x' y'$ ✓	$x' y'$
	$w' x' z'$ ✓	$x' z'$
	$x' y' z'$ ✓	
$w' x' y' z$ ✓	$x' y' z$ ✓	
$w' x' y z'$ ✓	$x' y z'$ ✓	
$w x' y' z'$ ✓	$w x' y'$ ✓	
	$w x' z'$ ✓	
$w x' y' z$ ✓	$w y' z$	
$w x' y z'$ ✓	$w y z'$	
$w x y z'$ ✓	$w x y$	
$w x y' z$ ✓	$w x z$	
$w x y z$ ✓		

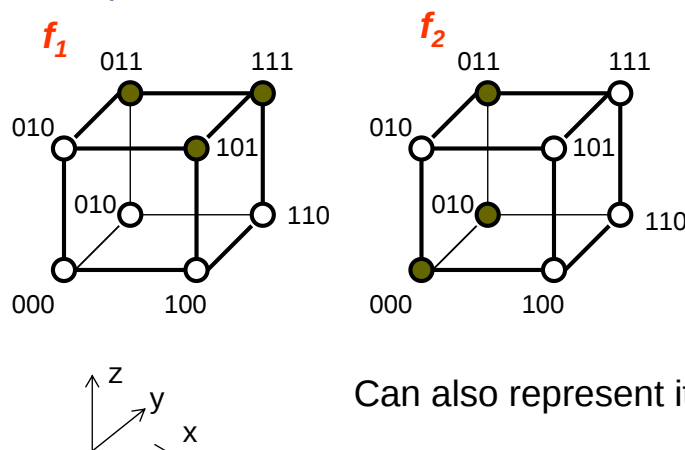
Courtesy: Maciej Ciesielski, UMass

15

Prime Generation for Multi-Output Function

- Similar to *single-output* function, except that we should include also the **primes of the products of individual functions**

■ Example



$x y z$	$f_1 f_2$
0-0	01
011	11
1-1	10

$x y z$	$f_1 f_2$
0-0	01
01-	01
-11	10
1-1	10

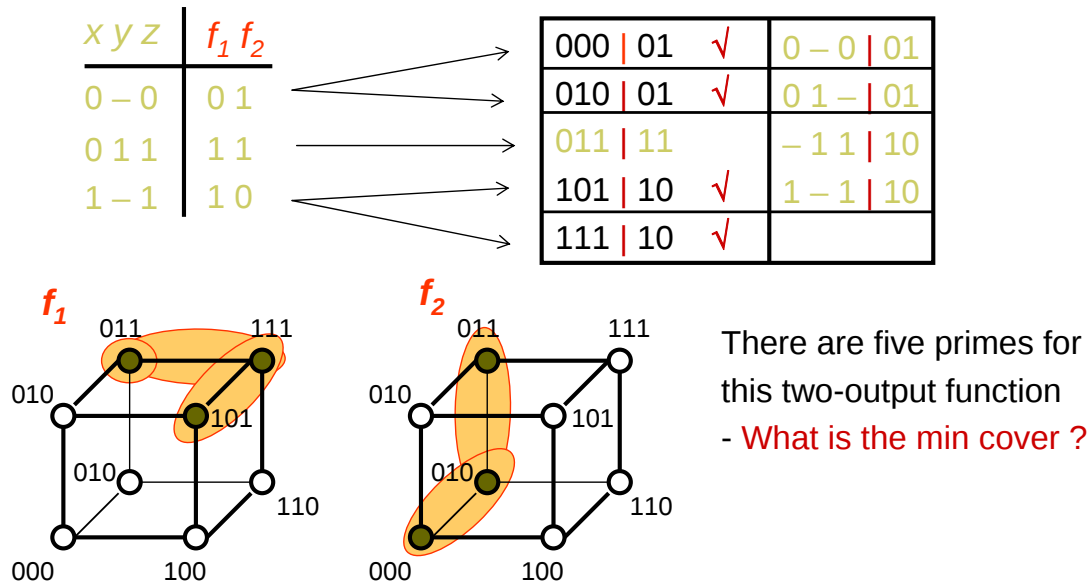
Can also represent it as:

16

Prime Generation

Example

- Modification from single-output case: When two adjacent implicants are merged, the output parts are **intersected**



17

Minimize Multi-Output Cover

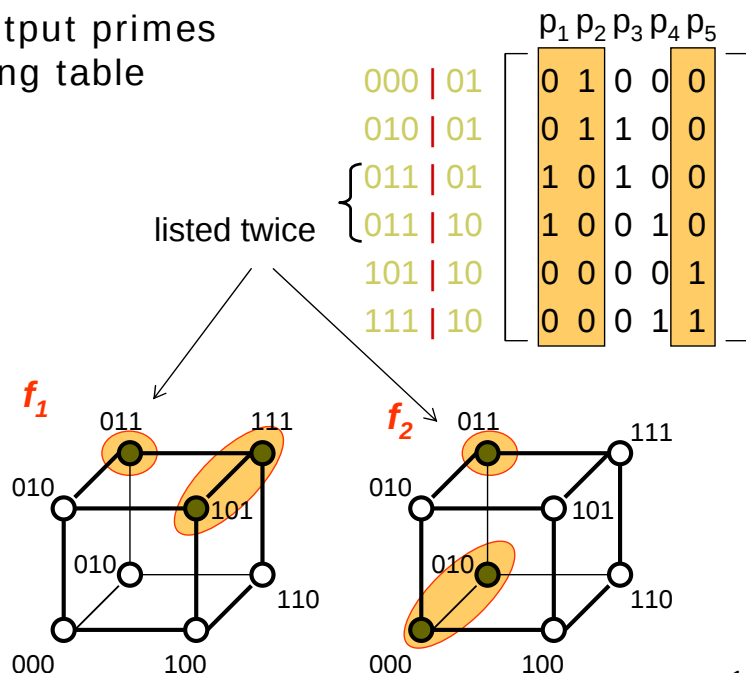
Example

- List multiple-output primes
- Create a covering table
- Solve

$$\begin{aligned}
 p_1 &= 011 | 11 \\
 p_2 &= 0-0 | 01 \\
 p_3 &= 01- | 01 \\
 p_4 &= -11 | 10 \\
 p_5 &= 1-1 | 10
 \end{aligned}$$

Min cover has 3 primes:

$$F = \{ p_1, p_2, p_5 \}$$



18

Prime Generation Using Unate Recursive Paradigm

- Apply **unate recursive paradigm** with the following **merge step**
 - (Assume we have just generated all primes of f_{x_i} and $f_{\neg x_i}$)
- **Theorem.**

p is a prime of f iff p is **maximal** (in terms of containment) among the set consisting of

 - $P = x_i q$, q is a prime of f_{x_i} , $q \not\subseteq f_{\neg x_i}$
 - $P = x_i' r$, r is a prime of $f_{\neg x_i}$, $r \not\subseteq f_{x_i}$
 - $P = q r$, q is a prime of f_{x_i} , r is a prime of $f_{\neg x_i}$

19

Prime Generation Using Unate Recursive Paradigm

- **Example**
 - Assume $q = abc$ is a prime of f_{x_i} . Form $p = x_i abc$.
 - Suppose $r = ab$ is a prime of $f_{\neg x_i}$. Then $x_i' ab$ is an implicant of f .

$$f = x_i abc + x_i' ab + abc + \dots$$

- Thus abc and $x_i' ab$ are implicants, so $x_i abc$ is not prime.
- **Note:** abc is prime because if not, $ab \subseteq f$ (or ac , or bc) contradicting abc prime of f_{x_i} .
- **Note:** $x_i' ab$ is prime, since if not then either $ab \subseteq f$, $x_i' a \subseteq f$, $x_i' b \subseteq f$. The first contradicts abc prime of f_{x_i} and the second and third contradict ab prime of $f_{\neg x_i}$.

20

Summary

□ Quine-McCluskey Method:

1. Generate cover of all primes $G = p_1 + p_2 + \dots + p_{3^n/n}$
2. Make G irredundant (in optimum way)

■ Q-M is **exact**, i.e., it gives an exact minimum

□ Heuristic Methods:

1. Generate (somehow) a cover of $\mathfrak{F}$ using some of the primes $G = p_{i_1} + p_{i_2} + \dots + p_{i_k}$
2. Make G irredundant (maybe not optimally)
3. Keep best result - try again (i.e. go to 1)