Logic Synthesis and Verification

Jie-Hong Roland Jiang
江介宏

Department of Electrical Engineering
National Taiwan University

Fall 2012

Two-Level Logic Minimization (1/2)

Reading:
Logic Synthesis in a Nutshell
Section 3 (§3.1-§3.2)

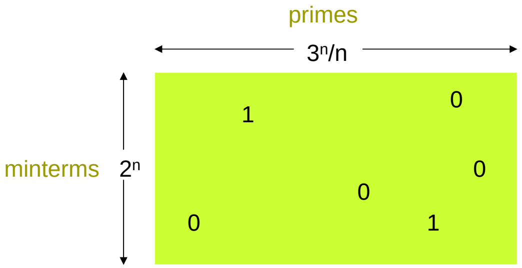
most of the following slides are by
courtesy of Andreas Kuehlmann

Quine-McCluskey Procedure

- Given G and D (covers for $\mathfrak{I}$ = (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 $\mathfrak{I}$)
- Q-M Procedure:
  1. Generate all primes of $\mathfrak{I}$, $\{P_j\}$ (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 if $m_i \in P_j$
     = 0 otherwise
  4. Solve the minimum column covering problem for B (unate covering problem)

Complexity

- ~2^n minterms; ~$3^n/n$ primes

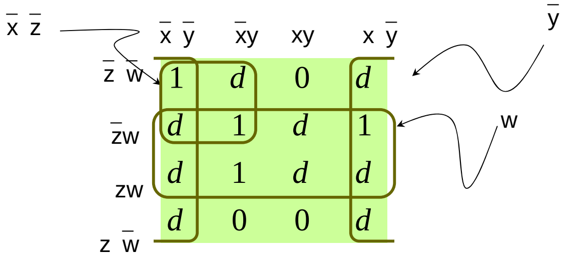


- There are $O(2^n)$ rows and $\Omega(3^n/n)$ columns. Moreover, minimum covering problem is NP-complete. (Hence the complexity can probably be double exponential in size of input, i.e. difficulty is $O(2^{3^n})$)

Two-Level Logic Minimization

- Example

Karnaugh map



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

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

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

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

Covering Table

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

Covering Table

Primes of f+d

Minterms of f

	$\bar{y}$	w	$\bar{x}\,\bar{z}$
$\bar{x}\,\bar{y}\,\bar{z}\,\bar{w}$	1	0	1
$\bar{x}y\,\bar{z}w$	0	1	1
$x\,\bar{y}\,\bar{z}w$	1	1	0
$\bar{x}yzw$	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.

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

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.

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.

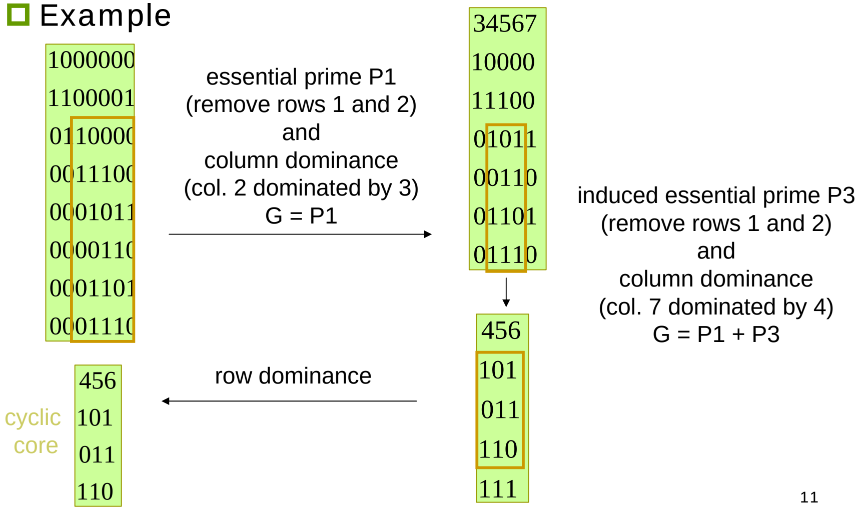
Covering Table
Table Reduction

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.

Covering Table
Table Reduction

- Example



1000000
1100001
0110000
0011100
0001011
0000110
0001101
0001110

essential prime P1
(remove rows 1 and 2)
and
column dominance
(col. 2 dominated by 3)
G = P1

34567
10000
11100
01011
00110
01101
01110

induced essential prime P3
(remove rows 1 and 2)
and
column dominance
(col. 7 dominated by 4)
G = P1 + P3

456
101
011
110
111

row dominance

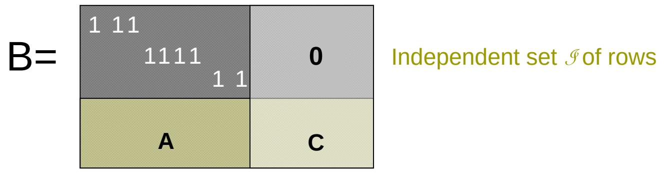
cyclic core

456
101
011
110

Solving Cyclic Core

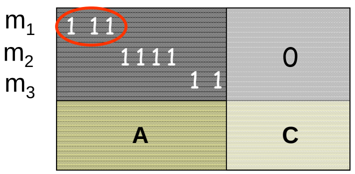
- Best known method (for unate covering) is branch and bound with some clever bounding heuristics
- Independent Set Heuristic:
  - Find a maximum set I of “independent” rows. Two rows B_{i1} ,B_{i2} are independent if **not** ∃j such that $B_{i_1j} = B_{i_2j} = 1$. (They have no column in common.)
  - Example

    A covering matrix B rearranged with independent sets first



Solving Cyclic Core

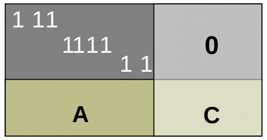
- **Lemma:**
  |Solution of Covering| ≥ $|\mathscr{I}|$


m_1 must be covered by one of the three columns

Solving Cyclic Core

- Heuristic algorithm:
  - Let $\mathscr{I} = \{I_1, I_2, \ldots, I_k\}$ be the independent set of rows

1. choose $j \in I_i$ such that column j covers the most rows of A. Put Pj in G
2. eliminate all rows covered by column j
3. $\mathscr{I} \leftarrow \mathscr{I} \setminus \{I_i\}$
4. go to 1 if $|\mathscr{I}| > 0$
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



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

$F = x'y' + wxy + x'yz' + wy'z$

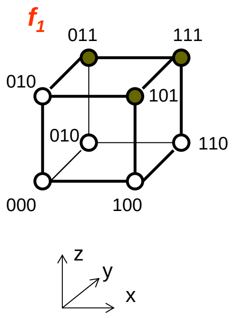
$F = x'y' + wxy + x'yz' + wy'z$

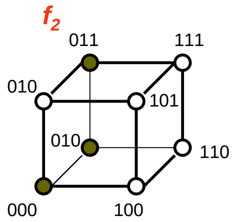
$w'x'y'z'$ √	$w'x'y'$ √ $w'x'z'$ √ $x'y'z'$ √	$x'y'$ $x'z'$
$w'x'y'z$ √ $w'x'yz'$ √ $wx'y'z'$ √	$x'y'z$ √ $x'yz'$ √ $wx'y'$ √ $wx'z'$ √	
$wx'y'z$ √ $wx'yz'$ √	$wy'z$ wyz'	
$wxyz'$ √ $wxy'z$ √	wxy wxz	
$wxyz$ √		

Courtesy: Maciej Ciesielski, UMASS

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





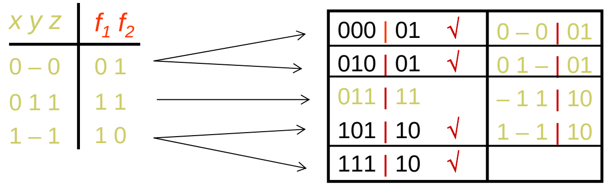
$x\,y\,z$	$f_1\,f_2$
0 – 0	0 1
0 1 1	1 1
1 – 1	1 0

Can also represent it as:

$x\,y\,z$	$f_1\,f_2$
0 – 0	0 1
0 1 –	0 1
– 1 1	1 0
1 – 1	1 0

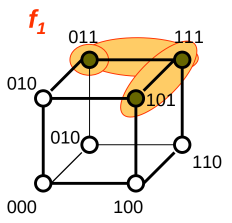
Prime Generation

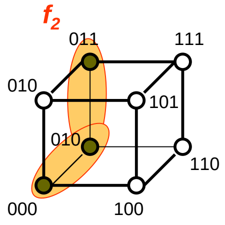
- Example
  - Modification from single-output case: When two adjacent implicants are merged, the output parts are intersected



$x\,y\,z$	$f_1\,f_2$
0 – 0	0 1
0 1 1	1 1
1 – 1	1 0

000 | 01 √	0 – 0 | 01
010 | 01 √	0 1 – | 01
011 | 11	– 1 1 | 10
101 | 10 √	1 – 1 | 10
111 | 10 √	





There are five primes for this two-output function

- What is the min cover ?

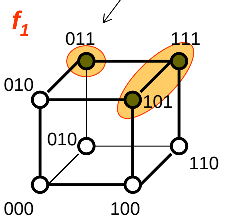
Minimize Multi-Output Cover

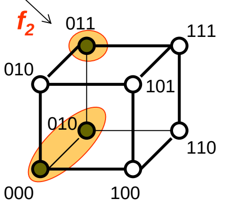
- Example
  1. List multiple-output primes
  2. Create a covering table
  3. Solve

$p_1 =$ 0 1 1 | 11
$p_2 =$ 0 – 0 | 01
$p_3 =$ 0 1 – | 01
$p_4 =$ – 1 1 | 10
$p_5 =$ 1 – 1 | 10

$$\begin{array}{r|ccccc|}
 & p_1 & p_2 & p_3 & p_4 & p_5 \\
000\,|\,01 & 0 & 1 & 0 & 0 & 0 \\
010\,|\,01 & 0 & 1 & 1 & 0 & 0 \\
011\,|\,01 & 1 & 0 & 1 & 0 & 0 \\
011\,|\,10 & 1 & 0 & 0 & 1 & 0 \\
101\,|\,10 & 0 & 0 & 0 & 0 & 1 \\
111\,|\,10 & 0 & 0 & 0 & 1 & 1
\end{array}$$

listed twice





Min cover has 3 primes:

F = { p_1, p_2, p_5 }

Prime Generation Using Unate Recursive Paragidm

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

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.

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

  - 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 $f_{\neg x_i}$.

Summary

- Quine-McCluskey Method:

1. Generate cover of all primes $G = p_1 + p_2 + \cdots + p_{3^n/n}$
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 $\mathfrak{I}$ using some of the primes $G = p_{i_1} + p_{i_2} + \cdots + p_{i_k}$
2. Make G irredundant (maybe not optimally)
3. Keep best result - try again (i.e. go to 1)