Two-Level Logic Synthesis -- Quine-McCluskey Method

Two-Level Logic

- What is Two-Level Logic?
- Why Two Levels?
  - Universal
  - Speed
  - Simplicity
- Typical Two Level Circuits:

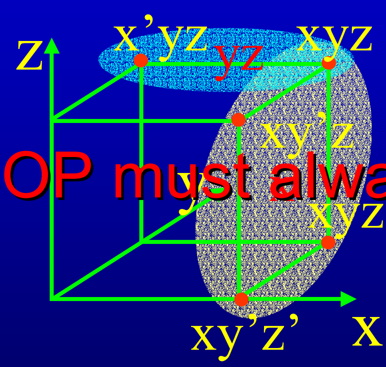
  AND-OR, OR-AND, NAND-NAND, NOR-NOR
- Cost Functions in Two-Level Circuits
  - Number of gates
  - Number of fanin (gate inputs)

Quine's PI Theorem

> Review: Implicants and PIs:

Let $f: B^n \rightarrow B$

- A minterm is an implicant if the corresponding discriminant is 1.
- An implicant with *k* literals has 2^{n-k} such minterms, who share these *k* literals.
- An implicant $l_1 \bullet\bullet\bullet l_k$ is prime iff $l_1 \bullet\bullet\bullet l_{j-1} l_{j+1} \bullet\bullet\bullet l_k$ is not an implicant for all $1 \leq j \leq k$.
- In the cubical representation, for an implicant with *k* literals,
  - $k = n$: a vertex (minterm)
  - $k = n-1$: an edge
  - $k = k$: an $(n-k)$-dimension face (plane)



> (Quine's Theorem [1952]) A minimal SOP must always consist of a sum of prime implicants.

> Minimal in terms of number of literals

Computing PIs: *Tabular Method*

$f(w,x,y,z) = x'y' + wxy + x'yz' + wy'z$

1. Rewrite in minterm canonical form;

   $f(w,x,y,z) = (wx'y'z+wx'y'z'+w'x'y'z+w'x'y'z')+$
   $(wxyz+wxyz')+(wx'yz'+w'x'yz')+(wxy'z+wx'y'z)$

2. Group by number of complement literals;
3. Merge terms in adjacent groups;

   Xy + Xy' = X -- distance-1 merging

4. Repeat step 3. until no new term is created.

$$f(w,x,y,z) = (wx'y'z+wx'y'z'+w'x'y'z+w'x'y'z')+(wxyz+wxyz')+(wx'yz'+w'x'yz')+(wxy'z)$$

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

Check for:

Xy + Xy' = X

All prime implicants:

wxy, wxz, wyz', wy'z, x'y', x'z'

More on Quine's Method:

- ↑ ***Easy to implement***
- ↑ **Can handle don't cares**
- ↓ **inefficient**

Computing PIs: Iterated Consensus Method

- Start with: SOP standard form (as compared to the canonical form in Tabular Method)
- Goal: sum of all PIs -- complete sum
- Basic idea: $xY + x'Z = xY + x'Z + YZ$ -- consensus law
- Theorem: A SOP formula is a complete sum iff
  - No term includes any other term;
  - The consensus of any two terms either does not exist or is contained in some term.

Example: *Iterated Consensus Method*

$f(w,x,y,z) = \underline{wx + x'y} + xyz$

$= wx + x'y + xyz + wy$

$= wx + \underline{x'y + xyz} + wy$

$= wx + x'y + xyz + wy + yz$

$= wx + x'y + wy + yz$

$xY + x'Z =$
$xY + x'Z + YZ$

- ↑ Start with any SOP form
- ↓ Need to compare every pair of terms
  - Do they have consensus?
  - Does one contain the other?

Computing PIs: Recursive Method

- Basic idea: if F_1 and F_2 are complete sums, the complete sum of $F_1 \cdot F_2$ can be obtained by:
  - Expanding F_1 and F_2 to POS (Boole's Expansion)
  - Multiplying out F_1 and F_2 by distributive law
  - Applying $x \cdot x = x$ and $x \cdot x' = 0$
  - Eliminating all terms that are contained in others
- Example: $f(w,x,y,z) = (w+x)(x'+y)(y+z)$

$$= (wx'+wy+\cancel{xx'}+xy)(y+z)$$

$$= wx'y+wx'z+wy\cancel{y}+wyz+xy\cancel{y}+xyz$$

$$= wy+xy+wx'y+wx'z+wyz+xyz$$

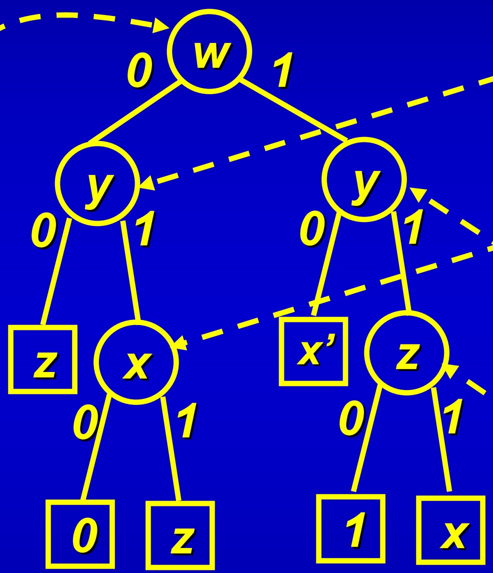
$$= wy+xy+wx'z$$

Computing the PIs: Recursive Method

- Given f(x,y,z….)= (x' + f(1,y,z..))(x + f(0,y,z…)) (Boolean Expansion)
- C-S(f) = ABSORB((x' +C-S(f(1,y,z..))(x + C-S(f(0,y,z…)))
- Essentially a recursive procedure. ABSORB is a manifestation of the result outlined in the previous slide.

Compute Complete Sum by Recursion Tree

$$f(w,x,y,z)=w'y'z+xyz+wyz'+wx'y'$$



$f(0,x,y,z)=y'z+xyz = (y+z)(y'+xz) = y'z+xz$

$f(0,x,0,z) = z$

$f(0,x,1,z) = xz = (x+0)(x'+z) = xz$

$f(0,0,1,z)=0$

$f(0,1,1,z)=z$

$f(1,x,y,z)= xyz+yz'+x'y' =(y+x')(y'+x+z')$

$=xy+yz'+x'y'+x'z'$

$f(1,x,0,z) = x'$

$f(1,x,1,z) = xz+z' = (z+1)(z'+x) = x+z'$

$f(1,x,1,0) =1$

$f(1,x,1,1) = x$

$f(w,x,y,z)$

$= (w+y'z+xz)(w'+xy+yz'+x'y'+x'z')$

$= wxy+wyz'+wx'y'+wx'z'+w'y'z+x'y'z+w'xz+xyz$

$f(x)= (x + f(0))$

$\bullet\ (x'+f(1))$

Quine-McCluskey Method

Problem: Given a Boolean function *f* (may be incomplete), find a minimum cost SOP formula.

↓ ***# of literals***

Q-M Procedure:

1. Generate all the PIs of *f*, $\{P_j\}$
2. Generate all the minterms of *f*, $\{m_i\}$
3. Build the Boolean constraint matrix B, where B_{ij} is 1 if $m_i \in P_j$ and is 0 otherwise
4. Solve the minimum column covering problem for B

Example: Quine-McCluskey Method

$$f(w,x,y,z) = x'y' + wxy + x'yz' + wy'z$$

	wxy	wxz	wyz'	$wy'z$	$x'y'$	$x'z'$
$wx'y'z'$					1	1
$w'x'y'z$					1	
$w'x'y'z'$					1	1
$wxyz$	1	1				
$wxyz'$	1		1			
$wx'yz'$			1			1
$w'x'yz'$						1
$wxy'z$		1		1		
$wx'y'z$				1	1	

minimum cover(s):

$\{x'y', x'z', wxy, wxz\}$,

$\{x'y', x'z', wxy, wy'z\}$,

$\{x'y', x'z', wxz, wyz'\}$.

More on Quine-McCluskey Method

- > Goal: find a minimum SOP form
- > Why We Need to Find all PIs?

$$f(w,x,y,z) = x'y' + wxy + x'yz' + wy'z$$

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

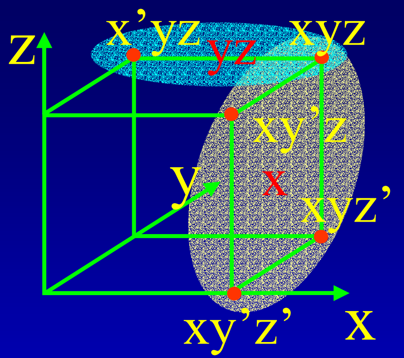
$$= x'y' + x'z' + wxy + wxz$$

$$= x'y' + x'z' + wxz + wyz'$$

1. *Are all terms PIs?*
2. *Is the form optimal?*
3. *Is the form unique?*

- > How We Find Them?
  - = Quine's tabular: start with minterm, the smallest I
  - = Iterated consensus: complete sum theorem
  - = Recursive: complete sum theorem

Quine's PI Theorem



> Review: Implicants and PIs:

Let f: $B^n \to B$

= A minterm is an implicant if the corresponding discriminant is 1.

= An implicant with k literals has 2^{n-k} such minterms, who share these k literals.

= An implicant $l_1 \bullet\bullet\bullet l_k$ is prime iff $l_1 \bullet\bullet\bullet l_{j-1} l_{j+1} \bullet\bullet\bullet l_k$ is not an implicant for all $1 \leq j \leq k$.

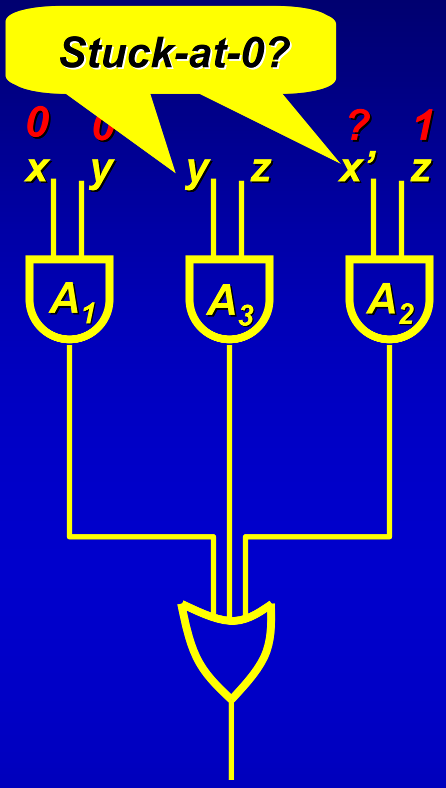
= In the cubical representation, for an implicant with k literals,

- $k = n$: a vertex (minterm)
- $k = n-1$: an edge
- $k = k$: an $(n-k)$-dimension face (plane)

> (Quine's Theorem [1952]) A minimal SOP must always consist of a sum of prime implicants.

Testability

- Testability
- Stuck-at-0/1 Fault Model
- Redundant Gate
- Untestable Fault



$A_3=1$: $y=1, z=1$
$A_1=0$: $x=0$
$A_2=x'z=1$

An area-optimal circuit must be fully testable for stuck-at faults.