ECE 474A/57A
Computer-Aided Logic Design

Logic Optimization 2

Qunie-McCluskey with Don't Cares, Iterated Consensus, Row/Column Dominance

K-map with Don't Cares

- Consider F(a, b, c) = Σm(2, 4) + Σd(1, 5, 6)
- What should we do with the don't cares?
  - Include d.c. if it helps to further minimize the cover (m5, m6)
  - Don't need to include d.c. if it doesn't help (better to exclude m1)

a \ bc	00	01	11	10
0	0	X	0	1
1	1	X	0	X

F = ab'c' + a'bc'

a \ bc	00	01	11	10
0	0	X	0	1
1	1	X	0	X

F = ab' + b'c + bc'

a \ bc	00	01	11	10
0	0	X	0	1
1	1	X	0	X

F = ab' + bc'

How do we apply these ideas to Quine-McCluskey?

Quine-McCluskey with Don't Cares
Example 1

- F(a, b, c) = Σm(2, 4) + Σd(1, 5, 6)

Step 1: Find all the prime implicants
- List all elements of on-set and don't care set, represented as a binary number
- Mark don't cares with "D"

G1	(1) 001 D	
	(2) 010	
	(4) 100	
G2	(5) 101 D	
	(6) 110 D	

Quine-McCluskey with Don't Cares
Example 1

Step 1: Find all the prime implicants (cont')
- Compare each entry in Gi to each entry in Gi+1
  - If they differ by 1 bit, we can apply the uniting theorem and eliminate a literal
  - If both values are don't cares, retain "D", otherwise no need to mark
  - Add check to implicant to remind us that it is not a prime implicant

G1	(1) 001 D ✓	G1	(1,5) -01 D	
	(2) 010 ✓		(2,6) -10	
	(4) 100 ✓		(4,5) 10-	
G2	(5) 101 D ✓		(4,6) 1-0	
	(6) 110 D ✓			

no new implicants are generated – end of step 1

we have found all prime implicants (ones without check marks)

Quine-McCluskey with Don't Cares

Example 1

Step 2: Create Prime Implicant Chart to find all essential prime implicants

- Minterms are added as columns in the table
- Prime implicants not marked as "D" are added as rows

Original Equation
F = Σm(2, 4) + Σd(1, 5, 6)

Derived in Step1
(1,5) -01 D
(2,6) -10
(4,5) 10-
(4,6) 1-0

	2	4
(2,6) P1		
(4,5) P2		
(4,6) P3		

Quine-McCluskey with Don't Cares

Example 1

Step 2: Create Prime Implicant Chart to find all essential prime implicants

- Place "X" in a row if the prime implicant covers the minterm
- Essential prime implicants are found by looking for rows with a single "X"
  - If minterm is covered by one and only one prime implicant – it's an essential prime implicant
- Add essential prime implicants to the cover

P1 is essential, need to include

Choose between P2 and P3 to cover remaining minterm

	2	4
(2,6) P1	✖	
(4,5) P2		✖
(4,6) P3		✖

Option 1
F = P1 + P2
F = bc' + ab'

Option 2
F = P1 + P3
F = bc' + ac'

Quine-McCluskey with Don't Cares

Example 2

- F = Σm(0, 3, 10, 15) + Σd(1, 2, 7, 8, 11, 14)

F ab \ cd	00	01	11	10
00	1	X	1	X
01			X	
11			1	X
10	X		X	1

Using a K-map we get

F = a'b' + ac

2 product terms, 2 variables each

Can we do just was well with Q.M.?

Quine-McCluskey with Don't Cares

Example 2

- F = Σm(0, 3, 10, 15) + Σd(1, 2, 7, 8, 11, 14)

G0	(0)	0000	
G1	(1)	0001	D
	(2)	0010	D
	(8)	1000	D
G2	(3)	0011	
	(10)	1010	
G3	(7)	0111	D
	(11)	1011	D
	(14)	1110	D
G4	(15)	1111	

G0	(0,1)	000-
	(0,2)	00-0
	(0,8)	-000
G1	(1,3)	00-1

combined on-set minterm and don't care, drop the "D"

combined two don't cares, keep the "D"

• • •

G0	(0,1,2,3)	00--
	(0,2,8,10)	-0-0
G1	(2,3,10,11)	-01-
G2	(3,7,11,15)	--11
	(10,11,14,15)	1-1-

These are your prime implicants (all smaller product terms have been combine into larger terms)

Quine-McCluskey with Don't Cares
Example 2

- F = Σm(0, 3, 10, 15) + Σd(1, 2, 7, 8, 11, 14)

		0	3	10	15
(0,1,2,3)	P1	✖	✖		
(0,2,8,10)	P2	✖		✖	
(2,3,10,11)	P3		✖	✖	
(3,7,11,15)	P4		✖		✖
(10,11,14,15)	P5			✖	✖

No essentials, how do we choose?
TRY PETRICKS!

F = (m0)(m3)(m10)(m15)
F = (P1+P2)(P1+P3+P4)(P2+P3+P5)(P4+P5)

Quine-McCluskey with Don't Cares
Example 2

F = $\underbrace{\text{(P1+P2)(P1+P3+P4)}}$(P2+P3+P5)(P4+P5)

P1P1 + P1P3 + P1P4 + P1P2 + P2P3 + P2P4
P1 × × ×

F = (P1+P2P3+P2P4)$\underbrace{\text{(P2+P3+P5)(P4+P5)}}$

P2P4 + P2P5 + P3P4 + P3P5 + P4P5 + P5P5
× × × P5

F = (P1+P2P3+P2P4)(P2P4+P3P4+P5)

F = P1P2P4+P1P3P4+P1P5+P2P2P3P4+P2P3P3P4+P2P3P5+P2P2P4P4+P2P3P4P4+P2P4P5
× × × P2P4 × ×

F = P1P3P4+P1P5+P2P3P5+P2P4

Best Options

Quine-McCluskey with Don't Cares
Example 2

- F = Σm(0, 3, 10, 15) + Σd(1, 2, 7, 8, 11, 14)

ab \ cd	00	01	11	10
00	1	X	1	X
01			X	
11			1	X
10	X		X	1

Original K-map

F = a'b' + ac

ab \ cd	00	01	11	10
00	1	X	1	X
01			X	
11			1	X
10	X		X	1

Q.M. Solution 1

F = P1P5
F = a'b' + ac

ab \ cd	00	01	11	10
00	1	X	1	X
01			X	
11			1	X
10	X		X	1

Q.M. Solution2

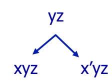
F=P2P4
F= b'd' + cd

Quine-McCluskey Overview

Quine-McCluskey Algorithm	**How is each step currently done?**	**Are there alternatives?**
(1) Find all prime implicants	**Tabular Minimization**	**Iterated Consensus to find complete sum**
(2) Find all essential prime implicants	**Prime Implicant Chart (column with single "X")**	**Constraint Matrix (basically same thing except axis switched)**
(3) Select a minimal set of remaining prime implicants that covers the on-set of the function	**Petrick's Method**	**Row/Column Dominance**

Iterated Consensus/Complete Sum

- Consider F(x, y, z) = yz + x′y + y′z′ + xyz + x′z′
- According to tabular minimization
  - First expanded product term into minterms
  - Then start comparing pairs to determine prime implicants



- Some of the work already done!
  - Instead we can take existing expression and determine the complete sum

Iterated Consensus/Complete Sum

Def: A complete sum is a SOP formula composed of all prime implicants of the function

Thm: A SOP formula is a complete sum if and only if

(1) No term includes any other term

(2) The consensus of any two terms of the formula either does not exist or is contained in some other term of the formula

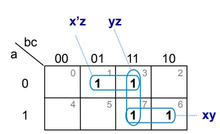
Iterated Consensus/Complete Sum

What is consensus?

In Boolean Algebra, consensus is defined as

(a) xy + x′z + yz = xy + x′z

(b) (x+y)(x′+z)(y+z) = (x+y)(x′+z)



K-map shows yz already covered by other two primes

Proof: xy + x′z + yz = xy + x′z

= xy + x′z + (x + x′)yz

= xy + x′z + xyz + x′yz

= (xy + xyz) + (x′z + x′yz)

= xy(1 + z) + x′z(1 + y)

= xy(1) + x′z(1)

= xy + x′z

Iterated Consensus/Complete Sum

- Typically consensus theorem used to simplify Boolean equations
  - Removed redundant terms

xy + x′z + yz = xy + x′z

(x+y)(x′+z)(y+z) = (x+y)(x′+z)

Ex $\underbrace{a}_{x}\underbrace{bc}_{y} + \underbrace{a'}_{x'}\underbrace{bd}_{z} + \underbrace{bcd}_{yz} = \underbrace{a}_{x}\underbrace{bc}_{y} + \underbrace{a'}_{x'}\underbrace{bd}_{z}$

Ex $abc'd + c'd'e + abc'e = c'(\underbrace{ab}_{y}\underbrace{d}_{x} + \underbrace{d'}_{x'}\underbrace{e}_{z} + \underbrace{abe}_{yz}) = c'(\underbrace{ab}_{y}\underbrace{d}_{x} + \underbrace{d}_{x'}\underbrace{e}_{z})$

Ex $(\underbrace{a}_{x}+\underbrace{b}_{y})(\underbrace{a'}_{x'}+\underbrace{c}_{z})(\underbrace{b+c}_{yz}) = (\underbrace{a}_{x}+\underbrace{b}_{y})(\underbrace{a'}_{x'}+\underbrace{c}_{z})$

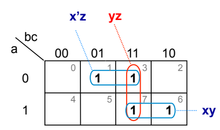
Ex (a+b)(c′+d)(a+c′) = cannot simplify

Iterated Consensus/Complete Sum

- We'll use consensus backwards
  - Add redundant terms

$\boxed{xy + x'z + yz = xy + x'z}$

- Generate complete sum
  - Is SOP not a complete sum, it's missing prime implicants
  - Missing prime must be covered by two or more implicants
  - Find the term spanning these implicants (consensus term), find the complete sum



K-map shows yz already covered by other two primes

F = xy + x'z // not a complete sum
F = xy + x'z + yz // a complete sum

- Why is it important to start with complete sum?
  - Better opportunity to apply absorption [x • (x + y) = x] with one of the original terms to obtain simpler expression
  - Step 1 of Quine McCluskey

Quine-McCluskey Overview

Quine-McCluskey Algorithm	How is each step currently done?	Are there alternatives?
(1) Find all prime implicants	Tabular Minimization	Iterated Consensus to find complete sum
(2) Find all essential prime implicants	Prime Implicant Chart (row with single "X")	Constraint Matrix (basically same thing except axis switched)
(3) Select a minimal set of remaining prime implicants that covers the on-set of the function	Petrick's Method	Row/Column Dominance

Iterated Consensus to Find Complete Sum

- Methodology to convert SOP function to complete sum

1. Start with arbitrary SOP form
2. Add consensus pair of all terms not contained in any other term
3. Compare new terms with existing and among other new terms to see if any new consensus terms can be generated
4. Remove all terms contained in some other term

Repeat 2 – 4 until no change occurs

Iterated Consensus to Find Complete Sum
Example 3

- F = yz + x'y + y'z' + xyz + x'z'

1. Start with arbitrary SOP form ✓

2. Add consensus pair of all terms not contained in any other term

yz + x'y = NO
yz + y'z' = yy' → 0, NO
yz + xyz = NO
yz + x'z' = x'y (INCL)

x'y + y'z' = x'z' (INCL)
x'y + xyz = yz (INCL)
x'y + x'z' = NO

y'z' + xyz = xzz' → 0, NO
y'z' + x'z' = NO

xyz + x'z' = xx'y → 0, NO

3. Compare new terms with existing and among other new terms to see if any new consensus terms can be generated

No new terms generated

4. Remove all terms contained in some other term

yz + x'y + y'z' + xyz + x'z'

yz + x'y + y'z' + x'z'

since there is a change you will need to start again – you will find in the next iteration no change occurs

Iterative vs. Recursive

- Iterative approach
  - Repetitive procedure used to add new consensus terms
- Recursive approach
  - Also repetitive, but we are trying to keep simplifying problem until solution is easy

Iterated Consensus Methodology

1. Start with arbitrary SOP form
2. Add consensus pair of all terms not contained in any other term
3. Compare new terms with existing and among other new terms to see if any new consensus terms can be generated
4. Remove all terms contained in some other term

Repeat 2 – 4 until no change occurs

Recursive Consensus Methodology

- Break down equation until it is trivial to find complete sum
  - Boole's expansion Theorem (a.k.a. Shannon Expansion)
  - Get down to 1 term, the complete sum of this term is itself

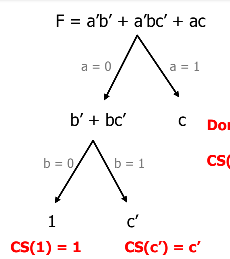
$$f(x_1, x_2, \ldots, x_n) = [x_1' \bullet f(0, x_2, \ldots, x_n)] + [x_1 \bullet f(1, x_2, \ldots, x_n)]$$
$$= [x_1' + f(1, x_2, \ldots, x_n)] \bullet [x_1 + f(0, x_2, \ldots, x_n)]$$

- Reconstruct equation, or equation's complete sum, using Thm 4.6.1 (Hatchel pg.138)

The SOP obtained from the two complete sums F1 and F2 by the following is a complete sum for F1 • F2

1. Multiply out F1 and F2 using the idempotent property (a+a=a, a•a=a), distributive properties, and x•x' = 0
2. Eliminate all terms contained in some other terms

Recursive Consensus Methodology
Example 4

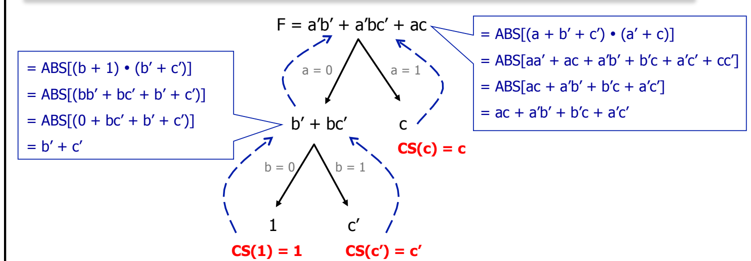


Done! We are down to 1 term

CS(c) = c

Now how do we put it all back together again?

Recursive Consensus Methodology
Example 4



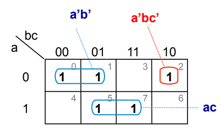
CS(F1 • F2) = ABS [CS(F1) • CS(F2)]

$$f(x_1, x_2, \ldots, x_n) = [x_1' + f(1, x_2, \ldots, x_n)] \bullet [x_1 + f(0, x_2, \ldots, x_n)]$$

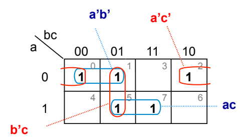
Recursive Consensus Methodology

Example 4

- Started with F = a'b' + a'bc' + ac
- Ended with CS(F) = ac + a'b' + b'c + a'c'
- Did it work?



Not a complete sum – missing some prime implicants



Complete sum achieved

Recursive method beneficial when dealing with larger equations

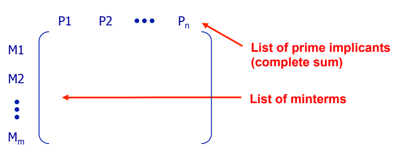
Book example F = v'xyz +v'w'x + v'x'z' + v'wxz + w'yz' + vw'z + vwx'z

Quine-McCluskey Overview

Quine-McCluskey Algorithm	How is each step currently done?	Are there alternatives?
(1) Find all prime implicants	Tabular Minimization	Iterated Consensus to find complete sum
(2) Find all essential prime implicants	Prime Implicant Chart (row with single "X")	Constraint Matrix (basically same thing except axis switched)
(3) Select a minimal set of remaining prime implicants that covers the on-set of the function	Petrick's Method	Row/Column Dominance

Constraint Matrix

- Describes conditions or constraints a cover must satisfy
  - Each column corresponds to a prime implicant
  - Each row correspond to a minterm



Note:
Similar to prime implicants chart. However, this textbook swaps the rows/cols

- GOAL – choose minimal subset of primes where each minter form which the function is 1 is included in at least one prime of the subset
  - Known as a "cover"

Constraint Matrix

Example 5

- F(x, y, z) = yz + x'y + y'z' + xyz + x'z'

Rows are minterms:

yz ⟶ **xyz, x'yz**
x'y ⟶ x'yz, **x'yz'**
y'z' ⟶ **xy'z', x'y'z'**
xyz ⟶ xyz (same)
x'z' ⟶ x'yz', x'y'z'

		P1 x'y	P2 x'z'	P3 y'z'	P4 yz
(m0)	x'y'z'	0	1	1	0
(m2)	x'yz'	1	1	0	0
(m3)	x'yz	1	0	0	1
(m7)	xyz	0	0	0	1
(m4)	xy'z'	0	0	1	0

Cols are prime implicants: *(get these from ex3)*

yz + x'y + y'z' + x'z'

Constraint Matrix
Example 5

- F(x, y, z) = yz + x'y + y'z' + xyz + x'z'

		P1 x'y	P2 x'z'	P3 y'z'	P4 yz
(m0)	x'y'z'	0	1	1	0
(m2)	x'yz'	1	1	0	0
(m3)	x'yz	1	0	0	1
(m7)	xyz	0	0	0	1
(m4)	xy'z'	0	0	1	0

Now we look for essential prime implicants

Singleton row (only one way to cover this minterm)
Must include these primes (essential) in the cover

		P1 x'y	P2 x'z'
(m2)	x'yz'	1	1

Remove P3 and P4 to simplify constraint matrix
Remove any minterm covered by these primes (m0, m3, m7, m4)

Easy to see how to cover remaining minterms (QM step 3)

Constraint Matrix
Example 5

- F(x, y, z) = yz + x'y + y'z' + xyz + x'z'

From previous slides …

P1	P2	P3	P4
x'y	x'z'	y'z'	yz

Solution 1
= P3 + P4 + P1
= y'z' + yz + x'y

Solution 2
= P3 + P4 + P2
= y'z' + yz + x'z'

What happens when the solution is not so obvious?

Quine-McCluskey Overview

Quine-McCluskey Algorithm	How is each step currently done?	Are there alternatives?
(1) Find all prime implicants	Tabular Minimization	Iterated Consensus to find complete sum
(2) Find all essential prime implicants	Prime Implicant Chart (row with single "X")	Constraint Matrix (basically same thing except axis switched)
(3) Select a minimal set of remaining prime implicants that covers the on-set of the function	Petrick's Method	Row/Column Dominance

Row Dominance (Constraint)

- If a row r_i in a constraint matrix has all the ones of another row r_j, we say r_i dominates r_j
- r_i is unneeded and all dominating row can be removed
  - Absorption property x • (x + y) = x

	P1	P2	P3
m1	1	1	0
m2	1	1	1

m2 dominates m1
remove m2

	P1	P2	P3
m1	1	1	0

Column Dominance (Variable)

- If column P_i has all the ones of another column P_j, and the cost of P_i is not greater than P_j, we say P_i dominates P_j
- The dominated column can be removed

	P1	P2	P3
m1	1	1	0
m2	0	1	1

P2 dominates P1
P2 dominates P3

Remove P1 and P3

	P2
m1	1
m2	1

- Assumes P2 does not cost more than P1 or P3
- What is the cost of a column?
  - Each prime implicant (col) corresponds to one AND gate
- Our Choices
  - We could say each column = 1 gate and everyone's the same
  - We could include number of literal, then a prime with 5 literals cost more than 3

Reduction Techniques Using Row/Col Dominance

1. Remove rows covered by "essential columns" (i.e. essential prime implicants)
2. Remove rows through row dominance (dominating row removed)
3. Remove columns through column dominance (dominated column removed)

Re-iterate 1-3 until no further simplification is possible

Reduction Techniques Using Row/Col Dominance
Example 6

	P1	P2	P3	P4	P5	P6
m1	1	1				1
m2	1	1				
m3		1	1			
m4		1	1	1		
m5				1	1	
m6				1	1	1

(1A) No essential columns to remove

	P1	P2	P3	P4	P5	P6
m2	1	1				
m3		1	1			
m5				1	1	

(2A) Row dominance
m1 dominates m2
m4 dominates m3
m6 dominates m5

Remove the **dominating** rows

Reduction Techniques Using Row/Col Dominance
Example 6

Note: P6 tell us nothing, you can remove to simplify if you want

	P1	P2	P3	P4	P5	P6
m2	1	1				
m3		1	1			
m5				1	1	

(3A) Column dominance
P2 dominates P1
P2 dominates P3
P4 dominates P5, vice versa

Remove the **dominated** cols

	P2	P4
m2	1	
m3	1	
m5		1

(1A) Essential Columns

Reduction Techniques Using Row/Col Dominance
Example 6

$$\begin{array}{c|cc} & P2 & P4 \\ \hline m2 & 1 & \\ m3 & 1 & \\ m5 & & 1 \end{array}$$

(1A) Essential Columns

P4 only column to cover m5

essential prime implicant = {P4}

P2 only column to cover m2, m3

essential prime implicant = {P4, P2}

$$\left[\quad\right]$$

Simplify matrix

Matrix empty – no further simplification possible

$\boxed{\text{Cover} = P2 + P4}$

Reduction Techniques Using Row/Col Dominance

- What happens when matrix cannot be simplified?
  - No rows left
    - We have a terminal case and solved the problem
  - Rows left
    - Problem is cyclic
    - Alternative techniques such as divide-and-conquer or branch-and-bound are needed
    - (Or guess, or use Petrick's)

Conclusion

- Quine-McCluskey with Don't Cares
- Alternative methods to perform Quine-McCluskey algorithm
- Iterated consensus (iterative and recursive)
- Generate a complete sum
- Row/Column Dominance