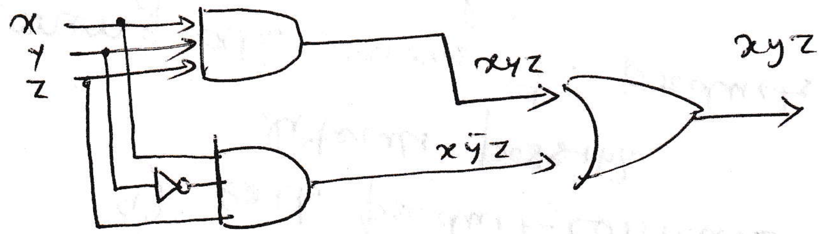
Week 14

Minimization of circuits

- The efficiency of a combinational circuit depends on the number and arrangements of its gates.
- The process of designing a combinational circuit
  - Begin with the table specifying the output for each combination of input values.
  - may use the sum-of-products expansion of the circuit & find a set of logic gates.
- The sum-of-product expansion may contain many more terms than are necessary.

Example:



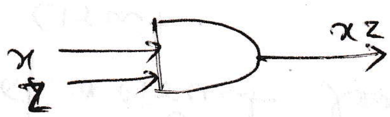
2 AND gate
1 OR gate
1 Inverter.

Sum-of-products expansion is $xyz + x\bar{y}z$

these terms differ only in one variable, and so can be combined.

$$xyz + x\bar{y}z = x(y+\bar{y})z$$
$$= x.1.z$$
$$= xz$$

→ a Boolean expression with fewer operators representing the circuit.



only one AND gate.

- Combining terms in the sum-of-products expansion of a circuit leads to a simpler expression for the circuit.

Minimization of the Boolean function

- produces Boolean sums of Boolean products that represent a Boolean fun. with the
  - a) fewest products of literals
  - a) each product contains the fewest literals possible.

  among all sums of products that represent the Boolean fun.

- makes it possible to construct a circuit for this Boolean fun. that uses
  - a) fewest gates
  - b) fewest inputs to the AND & OR gates ~~to~~.

  in the circuit, among all the Boolean expression we are minimizing.

- Karnaugh maps (or K-map), 1950.
  → useful in minimizing circuits upto six variables.
- Quine-McCluskey method, 1960
  → automates the process of minimizing combinational circuits, can be implemented as a computer program

K-map Method (Graphical), 1953, Karnaugh

- visual method for simplifying sum-of-products expansions
- applied only when the fun^n ~~has~~ involves six or fewer variables.

K-map with 2 variable → a square of 4 cells.

	y	$\bar{y}$
x	xy	$x\bar{y}$
$\bar{x}$	$\bar{x}y$	$\bar{x}\bar{y}$

- 4 possible minterms
- 4 cells ~~of~~ ~~4~~
- place 1 ~~B~~ in a cell i[f] the corresponding min-term represents the cell is in the given Boolean expression.

- adjacent cells have minterms that differ in exactly one literal.

Example: Find the K-map for

a) $xy + \bar{x}y$

1	
1	

b) $x\bar{y} + \bar{x}y$

	1
1	

c) $x\bar{y} + \bar{x}y + \bar{x}\bar{y}$

	1
1	1

- identify minterms that can be combined from the K-map.
- Is in two adjacent cells in the K-map, the 2 minterms can be combined to produce a product involving only one variable.

	y	
x	1	
$\bar{x}$	1	

$$xy + \bar{x}y = (x+\bar{x})y = y.$$

- 1's in all four cells, the four min-terms can be combined into one term, namely the Boolean expression 1 that involves none of the variables.

- Circle blocks of cells in the K-map that represent minterms that can be combined & then find the corresponding sum of products.

- goal is to identify the largest possible blocks, & cover all the 1s with the fewest blocks using the largest blocks ~~& always~~ first & always using the largest possible ~~B~~ blocks.

<u>Example:</u>

	y	$\bar{y}$
x	1	
$\bar{x}$	1	

a) $xy + \bar{x}y = y$

b) $x\bar{y} + \bar{x}y$

	y	$\bar{y}$
x		1
$\bar{x}$	1	

$\cancel{x\bar{y} + \bar{x}y}$

c) $x\bar{y} + \bar{x}y + \bar{x}\bar{y}$

$= \cancel{y +}\ \bar{x} + \bar{y}$

	y	$\bar{y}$
x		1
$\bar{x}$	1	1

A K-map with 3 variables

one way shown: larger blocks combined → fewer literals.

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x	1	2	4	3
$\bar{x}$	5	6	8	7

Think of it as lying on a cylinder.

Cell 1, Cell 3 adjacent
Cell 5, cell 7 adjacent. (two cells with common boundary are adjacent)

To simplify a sum-of-product expansion in 3 variables

- Use K-map to identify the blocks of minterms that can be combined.

- blocks of two adjacent cells represent pair of minterms that can be combined into a product of ~~single~~ 2 literal

- blocks of 2×2 & 4×1 cells represent minterms that can be combined into a single literal.

- blocks of all 8 cells represents a product of no literals.

· **Implicant**: The product of literals corresponding to a block of 1s in the K-map is called an implicant of the fun. to be minimized.

· **Prime implicant**: It is called prime implicant if this block of 1's is not contained in a larger block of 1s representing the product of fewer literals than in this product.

- Goal is to identify the largest possible blocks in the map & cover all the 1s in the map with the least no. of blocks, using largest blocks first.

Essential prime implicant: The largest blocks are always chosen, but we must always choose a block if it is the only block of 1s covering a 1 in the k-map. Such a block is called an essential prime implicant.

- Covering all the 1s in the map with blocks corresponding to prime implicants, we can express the sum-of-products as a sum of prime implicants.

Note- There may be more than one way to cover all the 1s using the least number of blocks.

Example: Use k-maps to minimize the sum-of-product expansions.

a) $xy\bar{z} + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}\bar{y}\bar{z}$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x		1		1
$\bar{x}$	1			1

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x		1	1	
$\bar{x}$	1		1	1

$x\bar{z} + \bar{x}yz + \cancel{\bar{x}\bar{y}z}$
$+ \bar{y}\bar{z}$

(b) $x\bar{y}z + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}\bar{y}z + \bar{x}\bar{y}\bar{z}$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x			1	1
$\bar{x}$	1		1	1

∴ minimal expansion into Boolean sums of Boolean products is

$\bar{y} + \bar{x}z$.

(c) $xyz + xy\bar{z} + x\bar{y}z + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}\bar{y}z + \bar{x}\bar{y}\bar{z}$.

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x	1	1	1	1
$\bar{x}$	1		1	1

Simplified minimal form is

$z + \bar{y} + x$

(d) $xy\bar{z} + x\bar{y}\bar{z} + \bar{x}\bar{y}z + \bar{x}\bar{y}\bar{z}$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x		1	1	
$\bar{x}$			1	1

Simplified minimal form is

$x\bar{z} + \bar{y}\bar{z} + \bar{x}\bar{y}$

essential prime implicants.

K-map with 4 variables

→ a square divided into 16 cells, representing

⇒ 16 possible minterms

One of the ways to form a K-map in four variables

w, x, y, z

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	•			✓
$w\bar{x}$				
$\bar{w}\bar{x}$				✓
$\bar{w}x$	✓		✓	•

→ two cells are adjacent iff the minterms they represent differ in one literal.

→ each cell is adjacent to four other cells.

→ This K-map can be thought of as lying on a torus so that adjacent cells have common boundary

Simplification procedure

- identify those blocks of 2, 4, 8 or 16 cells that represent minterms that can be combined.

- each cell representing a minterm must either be used to form a product using fewer literals, or be included in the expansion.

Example:

a) $wxyz + wxy\bar{z} + wx\bar{y}\bar{z} + w\bar{x}yz + w\bar{x}\bar{y}z$
$+ w\bar{x}\bar{y}\bar{z} + \bar{w}x\bar{y}z + \bar{w}\bar{x}yz + \bar{w}\bar{x}y\bar{z}$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	1	1	1	
$w\bar{x}$	1		1	1
$\bar{w}\bar{x}$	1	1		
$\bar{w}x$				1

$wx\bar{z} + w\bar{x}\bar{y} + \bar{w}\bar{x}y$
$+ wyz + \bar{w}x\bar{y}z$

5 implicants

least no. of blocks covering all 1's

or

1	1	1	
1		1	1
1	1		
			1

X

6 implicants.

not least # of blocks covering all 1's.

b) $\cancel{\bar y \bar z + w \bar x y + x \bar z}$

$w x \bar y \bar z + w \bar x y z + w \bar x y \bar z + w \bar x \bar y \bar z + \bar w x \bar y \bar z$

$+ \bar w \bar x y \bar z + \bar w \bar x \bar y \bar z$

	yz	$y\bar z$	$\bar y \bar z$	$\bar y z$
wx			1	
$w\bar x$	1	1	1	
$\bar w \bar x$		1	1	
$\bar w x$			1	

$\bar x \bar z + \bar y \bar z + w \bar x y$

c) $w x y \bar z + w x \bar y \bar z + w \bar x y z + w \bar x y \bar z + w \bar x \bar y \bar z$

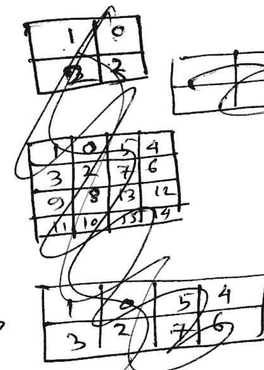
$+ \bar w x y z + \bar w x y \bar z + \bar w x \bar y \bar z + \bar w x \bar y z$

$+ \bar w \bar x y \bar z + \bar w \bar x \bar y \bar z$.

	yz	$y\bar z$	$\bar y \bar z$	$\bar y z$
wx		1	1	
$w\bar x$	1	1	1	
$\bar w \bar x$		1	1	
$\bar w x$	1	1	1	1

$\bar z + \bar w x + w \bar x y$

- Realistic with five or six variables.
- Beyond five or six variables, rarely used as K-map become extremely complicated.
- Concept is very useful to devise new algms to minimize Boolean fns. implemented in the Computer ~~adided~~ -aided design (CAD) programs



- Two variable → 2×2 rectangle,
- Three variable → 2×4 rectangle, use
- four variable → 4 × 4 rectangles

n variables → $2^{\lfloor \frac{n}{2} \rfloor} \times 2^{\lceil \frac{n}{2} \rceil}$

- Gray code used to assign a K-map to the minterms.

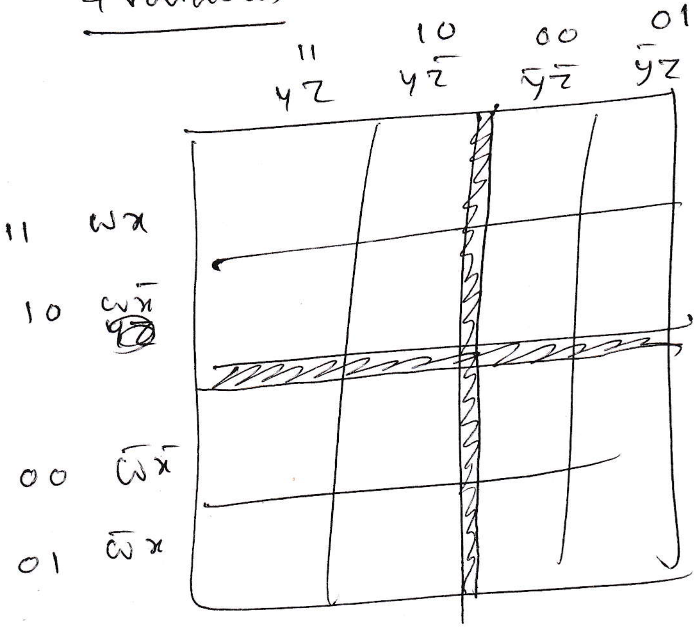
0 0   0
0 1   1
1 1
1 0

Taking reflexion.

2^4	2^3	2^2	2^1
1 1 1 1	1 1 1	1 1	1
1 1 1 0	1 1 0	1 0	0
1 1 0 0	1 0 0	mirror	mirror
1 1 0 1	1 0 1	0 0	
1 0 0 1	mirror	0 1	
1 0 0 0	0 0 1		
1 0 1 0	0 0 0		
1 0 1 1	0 1 0		
mirror	0 1 1		
0 0 1 1			
0 0 1 0			
0 0 0 0			
0 0 0 1			
0 1 0 1			
0 1 0 0			
0 1 1 0			
0 1 1 1			

Gray code construction
from length n to length n+1
a reflexion code.

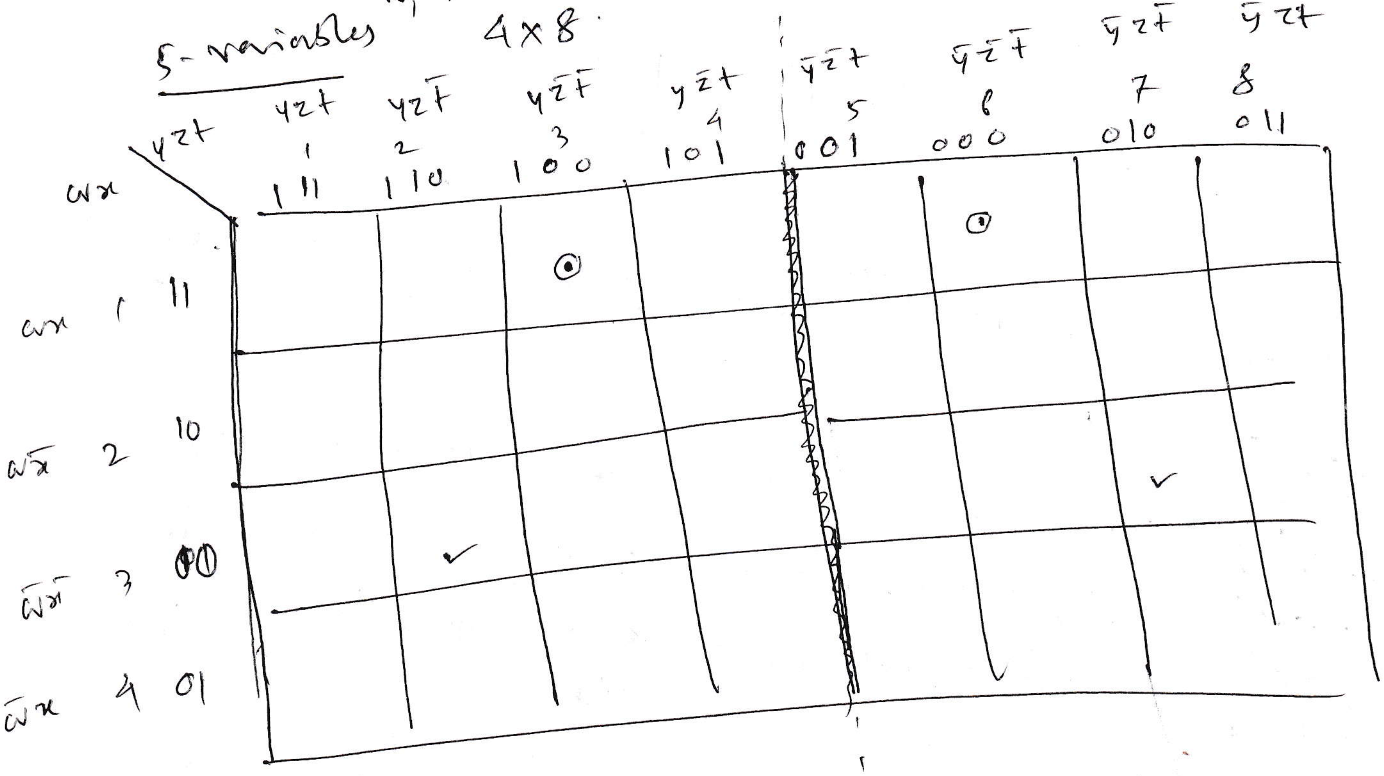
4 variables 4X4 w,x,y,z



Considering 1st. row & last row
cells adjacent & 1st. col., last col. cells adjacent
& using Gray Code to label rows & col's.
↓ ensures
minterms that differ in only one variable are always adjacent

Take the reflexion

5-variables w,x,y,z,t 4x8.



To ensure all cells representing products that differ in only one variable are considered adjacent

verify
→ Cells in top & bottom rows adjacent
→ 1st. & 8th col. adjacent
→ 1st. & 4th col. adjacent ; 2nd & 7th adjacent
→ 5th. & 8th. col. adjacent 3rd & 6th adjacent

13

Use of K-map to minimize a Boolean fun. in n variables

2^n cells, $2^{\lfloor n/2 \rfloor} \times 2^{\lceil n/2 \rceil}$

- Draw a K-map of the appropriate size.
  - K-map containing 2^n cells (i.e. n variables)
  - Use a rectangle containing $2^{\lfloor n/2 \rfloor}$ rows & $2^{\lceil n/2 \rceil}$ columns
  $\left(\lfloor \frac{n}{2} \rfloor + \lceil \frac{n}{2} \rceil = n \right)$.

- Place 1s in all cells corresponding to minterms in the sum-of-product expansion of this fun.

- Identify all prime implicants of the Boolean fun.
  - Look for the blocks ~~containing~~ consisting of 2^k clustered cells containing a 1, when $1 \leq k \leq n$.
  - These blocks correspond to the product of $n-k$ literals.
  - A block of 2^k cells each containing a 1 ~~not~~ not contained in a block of 2^{k+1} cells each containing 1
    → is a prime implicant.
    as
    no product obtained by deleting a literal is also represented by a block of cells all containing 1.

- Once all prime implicants have been identified, the goal is to find the smallest possible subset of these prime implicants with the property that: the cells representing these prime implicants cover all the cells containing a 1 in the K-map.

  - first select the essential prime implicants. as each of these is represented by a block that covers a cell containing a 1 that is not covered by any other prime implicant.

  - Add additional prime implicants to ensure that all 1's in the K-Map are covered.

    → This step can become exceedingly complicated if the # of variables is large.

Don't care conditions

- Some circuits care about only output for some combinations of input values
- Other combinations of input values are not possible or never occur.
- This gives us freedom in producing a simple circuit with the desired output
- The output values for all those combinations that never occur can be arbitrarily chosen.
- The values of fn. for these combinations are called don't care conditions.

• A d is used in k-map to mark those combinations of values of the ~~var~~ variables for which the fn. can be arbitrarily assigned.

• In the minimization process, we can assign 1s as values of those combinations of inputs values that lead to the largest blocks in the k-map.

16

Example: BCD (Binary Coded Decimal).

Code

Digit		BCD Codeword			F
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
D	w	x	y	z	F

BCD Code

873 ⟶ 1000 0111 0011

6 combinations out of 16 → never used to encode digits.

Q) Suppose that a circuit is to be built that produces an output of 1 if the decimal digit is ≥ 5
0 if the decimal digit is < 5.

How can this circuit be simply built using OR gate, AND gate, and inverters?

$$\therefore F = \bar{w}x\bar{y}z + \bar{w}xy\bar{z} + \bar{w}xyz + w\bar{x}\bar{y}\bar{z}$$
$$+ w\bar{x}\bar{y}z$$

The K-map for F, with d's in the don't care positions.

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	d	d	d	d
$w\bar{x}$	d	d	1	1
$\bar{w}\bar{x}$				
$\bar{w}x$	1	1		1

$F = w\bar{x}\bar{y} + \bar{w}xy + \bar{w}xz$.

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	d	d	d	d
$w\bar{x}$	d	d	1	1
$\bar{w}\bar{x}$				
$\bar{w}x$	1	1		1

$F = w\bar{x} + \bar{w}xy + x\bar{y}z$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	d	d	d	d
$w\bar{x}$	d	d	1	1
$\bar{w}\bar{x}$				
$\bar{w}x$	1	1		1

$wxyz$, $wx\bar{y}z$ — wxz — xz

$\bar{w}xyz$, $\bar{w}x\bar{y}z$ — $\bar{w}xz$

$wxz + \bar{w}xz$

$F = w + x\bar{y} + xz$

Simplest possible sum-of-product expansion.

Exercise: Simplify the Boolean fn. $F(w,x,y,z) = \Sigma(1,3,7,11,15)$ & the don't care conditions $d(w,x,y,z) = \Sigma(0,2,5)$.

Boolean product of sum simplification

Take minterms cor. to 0 → gives $\bar{F}$ → apply DeMorgan law.

- 1s placed in the sq's. of the map represent the minterm of F
- the minterms not included in F, denote the complement $\overline{F}$ of F.
- Complement $\overline{F}$ of F is represented in the map by a sq. not marked 1.
- mark empty sq's. by 0's & combine them into valid adjacent sq's.

→ get a simplified expression of $\overline{F}$ of F

- then apply DeMorgan's laws to $\overline{F}$ to get back F.

Example: $F = \sum(0, 1, 2, 5, 8, 9, 10)$

wx \ yz	00	01	11	10
00	1_{0}	1_{1}	$_{3}$	1_{2}
01	$_{4}$	1_{5}	$_{7}$	$_{6}$
11	$_{12}$	$_{13}$	$_{15}$	$_{14}$
10	1_{8}	1_{9}	$_{11}$	1_{10}

		0	
0		0	0
0	0	0	0
		0	

$\overline{F} = \overline{y}\overline{z} + \overline{w}\overline{x} + \overline{x}\overline{z}$

$F = (y+z)(w+x)(x+z)$

- Unfortunately, minimizing Boolean fun^{ns} with many variables is a computationally intensive problem
  → NP-complete problem.
  → No polynomial time algm. for minimizing Boolean circuits so far.

- Quine-McCluskey's method → exponential complexity.
  → ~~so~~ only used when # literals ≤ 10.

- So far, the best algms → minimizes only circuits with no more than 25 variables.

- Heuristic methods → used to substantially simplify Boolean expressions with a large no. of literals.
  → not necessarily minimiz

Example:

$F(w,x,y,z) = \sum(1,3,7,11,15)$

$d(w,x,y,z) = \sum(0,3,5)$

wx \ yz	00	01	11	10
00	d	1	1	d
01		d	1	
11			1	
10			1	

minimized fn. of F

$F = yz + \bar{w}\bar{x}$ — ①

wx \ yz	00	01	11	10
00	d	1	1	d
01		d	1	
11			1	
10			1	

minimized fn. of F

$F = yz + \bar{w}z$ — ②

$= z(\bar{w} + y)$

- ①,② not logically equivalent.
- min exp. of F not necessarily unique when don't care conditions are involved.

wx \ yz	00	01	11	10
00	d			d
01	0	d		0
11	0	0		0
10	0	0		0

minimized fn. of $\bar{F}$ (complement of F).

$\bar{F} = \bar{z} + w\bar{y}$

$\therefore F = \overline{(\bar{F})} = \overline{(\bar{z} + w\bar{y})}$

$= z\,(\bar{w} + y)$ — ③

②,③ are logically equivalent.

The Quine - McCluskey Method (1950).

- Can be used for ~~any~~ Boolean fcns. in any no. of variables.

- Two parts
  i) find those terms that are candidates for inclusion in a minimal expansion as a Boolean sum of Boolean products (dnf)
  ii) determine which of these terms to actually use.

Example Show how the Quine-McCluskey method can be used to find a minimal expansion equivalent to

$$xyz + x\bar{y}z + \bar{x}yz + \bar{x}\bar{y}z + \bar{x}\bar{y}\bar{z}$$

i)

	minterms	bitstring	# of 1's	
1.	xyz	111	3	(1,2)
2.	$x\bar{y}z$	101	2	(1,3)
3.	$\bar{x}yz$	011	2	(2,4)
4.	$\bar{x}\bar{y}z$	001	1	(3,4)
5.	$\bar{x}\bar{y}\bar{z}$	000	0	(4,5)

- minterms that can be combined are those that differ in exactly one literal.

- Hence, two terms that can be combined differ by exactly one in the # of 1s in the ~~bits~~ bit strings that represent them.

- When two minterms are combined into a product, this product contains two literals.

- A product in two literals is represented using a dash to denote the variable that does not occur.

Finding Boolean products with fewer variables

	Step 0			Step 1			Step 2	
	Terms	Bit strings		Term	Bit string		Term.	Bit String
1	xyz	111 (3)	(1,2)	xz	1–1	(1,2,3,4)	z	– –1
2	$x\bar{y}z$	101 (2)	(1,3)	yz	–11			
3	$\bar{x}yz$	011 (2)	(2,4)	$\bar{y}z$	–01			
4	$\bar{x}\bar{y}z$	001 (1)	(3,4)	$\bar{x}z$	0–1			
5	$\bar{x}\bar{y}\bar{z}$	000 (0)	(4,5)	$\bar{x}\bar{y}$	00–			

- Next, all pairs of products of two literals that can be combined are combined into one literals.

- Two such products can be combined if they contain literals for the same two variables, and literals for only one of the two variables differ.

ii) To identify a minimal set of products needed to represent the Boolean fun.

- Note that ~~The above~~ table indicate which terms have been used to form products with fewer literals (1,2,3,4). These terms will not be ~~used~~ needed in a minimal expansion.

- Begin with all those products that were not used to construct products with fewer literals.

Cover finding Table

22

	1	2	3	4	5
	$xy\bar{z}$	$x\bar{y}z$	$\bar{x}yz$	$\bar{x}\bar{y}z$	$\bar{x}\bar{y}\bar{z}$
z	X	X	X	X	
$\bar{x}\bar{y}$				X	X

Candidate product

- form a table which has a row for each candidate product formed by combining original terms & a column for each original term.
- put an X in a position if the original term in the sum-of-product expansion was used to form the candidate product.
- In this case, the candidate product is said to cover the original minterm. (e.g. $\bar{x}\bar{z}$ covers $\bar{x}\bar{y}z$ & $\bar{x}\bar{y}\bar{z}$)
- need to include at least one product that covers each of the original minterms.
- only one X in a column in the table → corresponding product in ~~the~~ the row of this X must be used.
- ~~For~~ from the above table, see that both z & $\bar{x}\bar{y}$ are needed.

Hence, the final answer is $z + \bar{x}\bar{y}$.

essential prime implicant

Sequence of Steps to Simplify a Sum-of-product expression in the Quine – McClusky method

1. express each minterm in n variables by a bit-string of length n while $\begin{cases} 1 & \text{in the } i\text{th position if } x_i \text{ occurs} \\ 0 & \text{in the } i\text{-th position if } \overline{x_i} \text{ occurs.} \end{cases}$

2. group the bit strings according to the # of 1's in them.

3. determine all products in $n-1$ variables that can be ~~combined~~ formed by taking the Boolean sum of minterms in the expansion.

   - minterms that can be combined are represented by bit strings that differ in exactly one positions.

   - represent these products ~~of~~ in $n-1$ variables with strings that have
     $\begin{cases} 1 & \text{in the } i\text{-th position if } x_i \text{ occurs in the product} \\ 0 & \text{in the } i\text{-th position if } \overline{x_i} \text{ occurs in the product} \\ - \text{(dash)} & \text{in the } i\text{-th position if there is no literal involving } x_i \text{ in product} \end{cases}$

4. Determine all products ~~of~~ in $n-2$ variables that can be ~~combined~~ formed by taking the Boolean sum of ~~minterms~~ products in $n-1$ variables found in the previous step.

- products in $n-1$ variables that can be combined are represented by bit strings that have a dash in the same position & differ in exactly one position.

5. Continue combining Boolean products into products with fewer variables as long as possible.

6. find all the Boolean products that arose that were not used to form a Boolean product in one fewer ~~of~~ literal.

7. Find the smallest set of these Boolean products such that the sum of these products represents the Boolean fun^n.

- done by forming a table ~~showth~~ showing which minterms are covered by which products.

- every minterm must be covered by at least one product.

- First find all essential prime implicants.

- ~~An~~ essential prime implicants is the only prime implicants that covers one of the minterms.

- Then simplify the table by eliminating the columns for minterms covered by this prime implicant

- eliminate any prime implicants that covers a subset of minterms covered by another prime implicant.

(e.g. $\bar{x}yz$ can be ignored while $w\bar{x}y$ is included ~~as~~, together with $\bar{w}z$ & $wy\bar{z}$ as:

~~as~~ $\bar{x}yz$ covers $\{w\bar{x}yz, \bar{w}\bar{x}yz\}$.

$\{2, 6\}$.

covered by $w\bar{x}y$

covered by $\bar{w}z$.

- eliminate column for a minterm if there is another minterm that is covered by a subset of the prime implicants that cover this minterm.

(e.g. minterm $\bar{w}\bar{x}yz$ (6) can be ignored.

covered by prime implicants $\underbrace{\bar{w}z \text{ \& } \bar{x}yz}$.

subset $\bar{w}z$ ~~Subset $\bar{w}z$~~ covers another minterm

$w\bar{x}y\bar{z}$ (3)

cover $\{wy\bar{z}, w\bar{x}y\}$.

subset ~~wy~~ $w\bar{x}y$ covers another minterm

Let minterm m_B & cover C

if $\exists$ another minterm m' with cover $C' \subseteq C$

delete column for m.

- This process of

  (a) identifying essential prime implicants. that must be included, followed by

  (b) eliminating redundant prime implicants and (eliminating dominated rows).

  (c) identifying minterms that can be ignored (eliminating dominating columns)

  is iterated untill the table does not change.

- (II) finally, use a backtracking procedure to find the optimal sol^n. when we add prime implicants to the cover to find possible sol^s. (finding cover is NP-complete problem).

27

Example: $wxy\bar{z} + w\bar{x}yz + w\bar{x}y\bar{z}$ ~~$+ \bar{w}\bar{x}y\bar{z}$~~ $+ \bar{w}xyz + \bar{w}x\bar{y}z + \bar{w}\bar{x}yz + \bar{w}\bar{x}\bar{y}z$.

	Step 0 Terms	Bit string		Step 1 Terms	Bitstring	Step 2 Term
1	$wxy\bar{z}$	1110 (3)	(1,3)	$wy\bar{z}$	1–10	(6,7,4,5) $\bar{w}z$
2	$w\bar{x}yz$	1011 (3)	×(1,5)			0––1
3	$w\bar{x}y\bar{z}$	1010 (2)	×(1,6)			
			(2,3)	$w\bar{x}y$	101–	
✓4	$\bar{w}xyz$	0111 (3)	×(2,5)			
✓5	$\bar{w}x\bar{y}z$	0101 (2)	(2,6)	$\bar{x}yz$	~~0~~–011	
✓6	$\bar{w}\bar{x}yz$	0011 (2)	✓(6,7)	$\bar{w}\bar{x}z$	✓00–1	
✓7	$\bar{w}\bar{x}\bar{y}z$	0001 (1)	×(4,3)			
			✓(4,5)	$\bar{w}xz$	✓01–1	
			✓(4,6)	$\bar{w}yz$	×0–11	
			(5,7)	$\bar{w}\bar{y}z$	×0–01	

prime implicants

essential prime implicants

	1 ✓ $wxy\bar{z}$	2 $w\bar{x}yz$	3 ✓ $w\bar{x}y\bar{z}$	4 ✓ $\bar{w}xyz$	5 ✓ $\bar{w}x\bar{y}z$	6 ✓ $\bar{w}\bar{x}yz$	7 ✓ $\bar{w}\bar{x}\bar{y}z$
$\bar{w}z$				X	X	X	X
$wy\bar{z}$	X		X				
$w\bar{x}y$		X	X				
$\bar{x}yz$		X				X	

$\bar{w}z + wy\bar{z} + w\bar{x}y$ or $\bar{w}z + wy\bar{z} + \bar{x}yz$ is the final answer.

28

The Tabulation procedure (Quine-McCluskey)

Example:

$$f(w,x,y,z) = \sum(0,1,2,5,7,8,9,10,13,15).$$

	minTerm	Bitstring	
1.	$\bar{w}\bar{x}\bar{y}\bar{z}$	0000	(0)
2.	$\bar{w}\bar{x}\bar{y}z$	0001	(1) ✓
3.	$\bar{w}\bar{x}y\bar{z}$	0010	(1) ✓
4.	$\bar{w}x\bar{y}z$	0101	(2) ✓
5.	$\bar{w}xyz$	0111	(3) ✓
6.	$w\bar{x}\bar{y}\bar{z}$	1000	(1) ✓
7.	$w\bar{x}\bar{y}z$	1001	(2) ✓
8.	$w\bar{x}y\bar{z}$	1010	(2) ✓
9.	$wx\bar{y}z$	1101	(3) ✓
10.	$wxyz$	1111	(4) ✓

	~~minTerm~~ Term	Bitstring	
1	$wxyz$	1111	(4)
2	$wx\bar{y}z$	1101	(3)
3	$\bar{w}xyz$	0111	(3)
4	$\bar{w}x\bar{y}z$	0101	(2)
5	$w\bar{x}\bar{y}z$	1001	(2)
6	$w\bar{x}y\bar{z}$	1010	(2)
7	$\bar{w}\bar{x}\bar{y}z$	0001	(1)
8	$\bar{w}\bar{x}y\bar{z}$	0010	(1)
9	$w\bar{x}\bar{y}\bar{z}$	1000	(1)
10	$\bar{w}\bar{x}\bar{y}\bar{z}$	0000	(0)

~~minterm~~	Term ~~Bitstring~~
~~(1,3)~~ (1,2)	wxz ✓
(1,3)	xyz ✓
(2,4)	$x\bar{y}z$ ✓
(2,5)	$w\bar{y}z$ ✓
(2,6)	w
(3,4)	$\bar{w}xz$ ✓
(3,5)	z
(3,6)	y
(4,7)	$\bar{w}\bar{y}z$ ✓
(4,8)	$\bar{w}$
(4,9)	y

(6,7) $\bar{x}$

(6,8) $\bar{x}y\bar{z}$ ✓

(6,9) $w\bar{x}\bar{z}$

✓ $\bar{x}\bar{y}z$ (5,7), $\bar{x}$ (5,8), $w\bar{x}\bar{y}$ (5,9), $\bar{w}\bar{x}\bar{y}$ ✓ (7,10), $\bar{w}\bar{x}\bar{z}$ ✓ (8,10), $\bar{x}\bar{y}\bar{z}$ ✓ (9,10) step1

	terms	Bitstring
1.	$wxyz$	1111 (4)
2.	$wx\bar{y}z$	1101 (3)
3.	$\bar{w}xyz$	0111 (3)
4.	$\bar{w}x\bar{y}z$	0101 (2)
5.	$w\bar{x}\bar{y}z$	1001 (2)
6.	$w\bar{x}y\bar{z}$	1010 (2)
7.	$\bar{w}\bar{x}\bar{y}z$	0001 (1)
8.	$\bar{w}\bar{x}y\bar{z}$	0010 (1)
9.	$w\bar{x}\bar{y}\bar{z}$	1000 (1)
10.	$\bar{w}\bar{x}\bar{y}\bar{z}$	0000 (0)

step2 (29)

	term	Bitstring
(1,2)	~~xyz~~ wxz	11–1 ~~((1,3), (2,4))~~
(1,3)	xyz	✓–111
(2,4)	$x\bar{y}z$	✓–101
(2,5)	$w\bar{y}z$	1–01 ✓
(3,4)	$\bar{w}x\bar{z}$	01–0
(4,7)	$\bar{w}\bar{y}z$	0–01 ✓
(5,7)	$\bar{x}\bar{y}z$	–001
(5,9)	$w\bar{x}\bar{y}$	100–
(7,10)	$\bar{w}\bar{x}\bar{y}$	000–
(8,10)	$\bar{w}\bar{x}\bar{z}$	00–0
(9,10)	$\bar{x}\bar{y}\bar{z}$	–000
(6,8)	$\bar{x}y\bar{z}$	–010
(6,9)	$w\bar{x}\bar{z}$	10–0

We obtain two minimal expressions for f_1 namely $f_2(w,x,y,z) = xz + \bar{x}\bar{z} + \bar{y}z$ and $f_2(w,x,y,z) = xz + \bar{x}\bar{z} + \bar{x}\bar{y}$

	5		7	
$\bar{y}z$	x		x	
$\bar{x}\bar{y}$	x			

(1,3)
(2,4)

	term	Bitstring
(1,3), (2,4)	xz	–1–1
(2,5), (4,7)	$\bar{y}z$	––01
(5,7), (9,10)	$\bar{x}\bar{y}$	–00–
~~(5,9), (7,10)~~	~~$\bar{x}\bar{y}$~~	
(8,10), (6,9)	$\bar{x}\bar{z}$	–0–0

prime implicants
$\rightarrow xz, \bar{y}z, \bar{x}\bar{y}, \bar{x}\bar{z}$.

5 dominates 7 can be deleted.

finding cover

	1 $wxyz$	2 $wx\bar{y}z$	3 $\bar{w}xyz$	4 $\bar{w}x\bar{y}z$	5 $w\bar{x}\bar{y}z$	6 $w\bar{x}y\bar{z}$	7 $\bar{w}\bar{x}\bar{y}z$	8 $\bar{w}\bar{x}y\bar{z}$	9	10
✓ xz	x	x	x	x						
$\bar{y}z$		x		x	x		x			
$\bar{x}\bar{y}$					x		x		x	x
✓ $\bar{x}\bar{z}$						x		x	x	x

Example: prime implicant chart.

$\cancel{f(w,x,y,z)} = \sum(1,3,4,5,6,7,10,11,12,13,14,15,18,19,20,21,22,23,25,26,27)$

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

	1 ✓	3 ✓	4 ✓	5 ✓	6 ✓	7 ✓	10	11	12 ✓	13 ✓	14 ✓	15 ✓	18	19	20 ✓	21 ✓	22 ✓	23 ✓	25 ✓	26	27 ✓
✓ $\bar{w}x$			x	x	x	x									x	x	x	x			
✓ $\bar{v}x$			x	x	x	x			x	x	x	x									
$v\bar{x}y$													x	x						x	x
$v\bar{w}y$													x	x			x	x			
$w\bar{x}y$							x	x												x	x
$\bar{v}wy$							x	x			x	x									
$\bar{x}yz$		x						x						x				x			x
$\bar{w}yz$		x				x								x				x			
$\bar{v}yz$		x				x		x				x									
✓ $\bar{v}\bar{w}z$	x	x		x		x															
✓ $vw\bar{x}z$																			x		x

as it covers a subset of minterms covered by $v\bar{x}y$

dominates → can be deleted

dominates → can be deleted

	10	11	18	19	26
✓ $v\bar{x}y$			x	x	x
no need $v\bar{w}y$			x	x	
✓ $w\bar{x}y$	x	x			x
no need $\bar{v}wy$	x	x			
✓ $\bar{x}yz$		x		x	x
no need $\bar{w}yz$				x	
no need $\bar{v}yz$		x			

dominates

can be deleted.

Reduced prime implicant chart.

	10 ✓	11 ✓	18 ✓	19 ✓	26 ✓
✓ $v\bar{x}y$			x	x	x
✓ $w\bar{x}y$	x	x			x
$\bar{x}yz$		x		x	

final chart.

10	18	26
	x	x
x		x

$\cancel{f(w,x,y.}$

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

$= \bar{w}x + \bar{v}x + \bar{v}\bar{w}z + vw\bar{x}z + v\bar{x}y + w\bar{x}y$.

Reducing # of rows:

- A row i of a prime ~~implication~~ implicant chart is said to dominate another row j of that chart if i covers every column covered by j.

- if row i dominates row j and the prime implicant corresponding to row i doesn't have more literals than ~~that~~ the prime implicant corresponding to row j, then row j can be deleted from the chart [i.e. eliminating the prime implicant (cor. to row j) that covers a subset of minterms covered by another prime implicant (cor. to row i).]

Reducing # of columns

- A column i in a prime implicant chart is said to dominate another column j of that chart if i has an x in every row in which j has an x. (i more x that j)

- if column i dominates column j, then column i can be deleted from the chart without affecting the search for a minimal expression.

Note-
- When reducing columns, the dominating ones are removed.
- When reducing rows, the dominated ~~rows~~ ones are removed.