Unit 2 – Optimized Implementation of Logic Functions

OVERVIEW

- We can always minimize logic functions using the Boolean theorems. However, more powerful methods such as Karnaugh maps and Quine-McCluskey algorithm exist: they provide a systematic way to arrive at the minimal form of a Boolean function.

KARNAUGH MAPS

2 VARIABLES:

x	y	f
0	0	m_0
0	1	m_1
1	0	m_2
1	1	m_3

y \ x	0	1	
0	m_0	m_2	$\bar{y}$
1	m_1	m_3	y
	$\bar{x}$	x	

x	y	f
0	0	0
0	1	1
1	0	1
1	1	0

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

f = x'y + y'x = m_1+m_2

x	y	f
0	0	1
0	1	1
1	0	0
1	1	1

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

f = x' + y

3 VARIABLES:

x	y	z	f
0	0	0	m_0
0	0	1	m_1
0	1	0	m_2
0	1	1	m_3
1	0	0	m_4
1	0	1	m_5
1	1	0	m_6
1	1	1	m_7

z \ xy	00	01	11	10	
0	m_0	m_2	m_6	m_4	$\bar{z}$
1	m_1	m_3	m_7	m_5	z
	$\bar{x}\bar{y}$	$\bar{x}y$	xy	$x\bar{y}$	

x	y	z	f
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

z \ xy	00	01	11	10	
0	1	0	0	1	$\bar{z}$
1	1	0	1	0	z
	$\bar{x}\bar{y}$	$\bar{x}y$	xy	$x\bar{y}$	

f = x'y' + z'y' + xyz

z \ xy	00	01	11	10	
0	1	1	1	1	$\bar{z}$
1	0	0	0	1	z
	$\bar{x}\bar{y}$	$\bar{x}y$	xy	$x\bar{y}$	

f = xy' + z'

z \ xy	00	01	11	10	
0	1	1	1	1	$\bar{z}$
1	1	0	0	1	z
	$\bar{x}\bar{y}$	$\bar{x}y$	xy	$x\bar{y}$	

f = y' + z'

4 VARIABLES:

x	y	z	w	f
0	0	0	0	m_0
0	0	0	1	m_1
0	0	1	0	m_2
0	0	1	1	m_3
0	1	0	0	m_4
0	1	0	1	m_5
0	1	1	0	m_6
0	1	1	1	m_7
1	0	0	0	m_8
1	0	0	1	m_9
1	0	1	0	m_{10}
1	0	1	1	m_{11}
1	1	0	0	m_{12}
1	1	0	1	m_{13}
1	1	1	0	m_{14}
1	1	1	1	m_{15}

zw \ xy	00	01	11	10	
00	m_0	m_4	m_{12}	m_8	$\bar{z}\,\bar{w}$
01	m_1	m_5	m_{13}	m_9	$\bar{z}\,w$
11	m_3	m_7	m_{15}	m_{11}	$z\,w$
10	m_2	m_6	m_{14}	m_{10}	$z\,\bar{w}$
	$\bar{x}\,\bar{y}$	$\bar{x}\,y$	$x\,y$	$x\,\bar{y}$	

zw \ xy	00	01	11	10	
00	1	0	0	1	$\bar{z}\,\bar{w}$
01	0	1	1	0	$\bar{z}\,w$
11	0	1	1	0	$z\,w$
10	1	0	0	1	$z\,\bar{w}$
	$\bar{x}\,\bar{y}$	$\bar{x}\,y$	$x\,y$	$x\,\bar{y}$	

f = y'w' + wy

x	y	z	w	f
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

zw \ xy	00	01	11	10	
00	1	0	0	0	$\bar{z}\,\bar{w}$
01	0	1	0	1	$\bar{z}\,w$
11	0	1	0	0	$z\,w$
10	1	0	0	0	$z\,\bar{w}$
	$\bar{x}\,\bar{y}$	$\bar{x}\,y$	$x\,y$	$x\,\bar{y}$	

f = x'y'w' + x'yw + xy'z'w

zw \ xy	00	01	11	10	
00	1	1	1	1	$\bar{z}\,\bar{w}$
01	0	0	0	0	$\bar{z}\,w$
11	1	0	0	1	$z\,w$
10	1	1	1	1	$z\,\bar{w}$
	$\bar{x}\,\bar{y}$	$\bar{x}\,y$	$x\,y$	$x\,\bar{y}$	

f = y'z + w'

DON'T CARE OUTPUTS

zw \ xy	00	01	11	10	
00	0	0	0	0	$\bar{z}\,\bar{w}$
01	1	1	1	X	$\bar{z}\,w$
11	0	0	0	0	$z\,w$
10	0	0	X	0	$z\,\bar{w}$
	$\bar{x}\,\bar{y}$	$\bar{x}\,y$	$x\,y$	$x\,\bar{y}$	

f = z'w

zw \ xy	00	01	11	10	
00	0	0	0	0	$\bar{z}\,\bar{w}$
01	1	0	0	1	$\bar{z}\,w$
11	1	0	0	1	$z\,w$
10	0	0	X	0	$z\,\bar{w}$
	$\bar{x}\,\bar{y}$	$\bar{x}\,y$	$x\,y$	$x\,\bar{y}$	

f = wy'

- This method appears in: "The Map Method for Synthesis of Combinational Logic Circuits", Maurice Karnaugh, *Transactions of the AIEE, Part I: Communication and Electronics*, vol. 72, no. 5, Nov. 1953, pp. 593-599. Karnaugh maps of 5, 6, 7, 8, and 9 are hinted at. Beyond 9 variables, the mental gymnastics for minimization become formidable.
- The Quine-McCluskey algorithm provides a simpler approach when dealing with a relatively large number of variables.

QUINE-MCCLUSKEY ALGORITHM

- This method appears in: "Minimization of Boolean Functions", E. J. McCluskey, Jr., *The Bell System Technical Journal*, vol. 35, no. 6, Nov. 1956, pp. 1417-1444.
- **Literal**: For an $n-$variable function F, it is a variable expressed as X or $\overline{X}$.
- **Implicant**: For an $n-$variable function, it is any product term that can appear in any possible sum of products (canonical or non-canonical) that represents the function. If P is an implicant, then $P = 1$ implies that the function is 1. Thus, every minterm is an implicant.
  A graphical way to see the implicants of a function is to take a look at the Karnaugh map (for a relatively low number of variables). All the possible terms we can get out of the K-map are implicants.
- **Prime implicant:** It is an implicant P such that the removal of any literal from P results in non-implicant of the function.

OUTLINE

1. **Get the function to be minimized represented as a canonical Sum of Products:** Use the minterm expansion form.

$$F(A,B,C,D) = \sum m(0,1,2,5,6,7,8,9,10,14)$$

2. **Get the Prime Implicants of the function:** This is done by systematically applying $XY + X\overline{Y} = X$ to all possible minterms and resulting non-canonical product terms. So, we build the Implicants Table by determining all Implicants:
   - ✓ We represent the minterms using the binary notation. For example: $m_1 = \overline{A}\overline{B}\overline{C}D = 0001$. Then, we group the minterms by the number of ones they contain. For an $n-$variable function, the minterms have n literals.
   - ✓ We apply $XY + X\overline{Y} = X$ to all possible pairs of minterms. This applies to pair of minterms that only vary by one literal. We attach a '✓' to every minterm that was employed.

$$m_{0,1} = m_0 + m_1 = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D = \overline{A}\overline{B}\overline{C}$$

   Note the table representation: $m_{0,1} = m_0 + m_1 = 0000 + 0001 = 000-$. The symbol "$-$" indicates that a literal was simplified. The resulting column consists of terms with $n-1$ literals.
   - ✓ We keep applying $XY + X\overline{Y} = X$ to all possible pair of resulting product terms. We attach a '✓' to every term that was employed. For each column we add, an extra literal is simplified (or a symbol "$-$" is added to the terms).

$$m_{0,1,8,9} = m_{0,1} + m_{8.9} = \overline{A}\overline{B}\overline{C} + A\overline{B}\overline{C} = \overline{B}\overline{C} \equiv 000- \ + 100- = -00-$$

   If we happen to get a repeated term, we eliminate one:

$$m_{0,1,8,9} = m_{0,8,1,9} = -00-, \qquad \rightarrow m_{0,8,1,9}\ is\ eliminated$$

   - ✓ When we cannot simplify any further, we stop and look for the terms that do not have a check '✓'. These terms are called the **Prime Implicants**. All the terms that appear in the table are the **Implicants**.

Number of ones	4-literal implicants	3-literal implicants	2-literal implicants	1-literal implicants
0	m_0 = 0000 ✓	$m_{0,1}$ = 000- ✓ $m_{0,2}$ = 00-0 ✓ $m_{0,8}$ = -000 ✓	**$m_{0,1,8,9}$ = -00-** **$m_{0,2,8,10}$ = -0-0** ~~$m_{0,8,1,9}$ = -00-~~	We can't combine any further, so we stop here
1	m_1 = 0001 ✓ m_2 = 0010 ✓ m_8 = 1000 ✓	**$m_{1,5}$ = 0-01** $m_{1,9}$ = -001 ✓ $m_{2,6}$ = 0-10 ✓ $m_{8,9}$ = 100- ✓ $m_{8,10}$ = 10-0 ✓ $m_{2,10}$ = -010 ✓	**$m_{2,6,10,14}$ = --10** ~~$m_{2,10,6,14}$ = --10~~	
2	m_5 = 0101 ✓ m_6 = 0110 ✓ m_9 = 1001 ✓ m_{10}= 1010 ✓	**$m_{5,7}$ = 01-1** **$m_{6,7}$ = 011-** $m_{6,14}$ = -110 ✓ $m_{10,14}$ = 1-10 ✓		
3	m_7 = 0111 ✓ m_{14}= 1110 ✓			
4				

$$F(A,B,C,D) = \overline{A}\overline{C}D + \overline{A}BD + \overline{A}BC + \overline{B}\overline{C} + \overline{B}\overline{D} + C\overline{D}$$

3. **Select a minimum set of Prime Implicants**: F is the sum of this set that contains the minimum number of literals.
   - ✓ Build the Prime Implicant Chart. Mark the minterms that cover each single Prime Implicant with an 'X'.
   - ✓ Get the **Essential Prime Implicants**: Look for minterms that are covered by (are part of) a single Prime Implicant: this is, look for columns with one X. The corresponding Prime Implicants are the Essential Prime Implicants.
     The minimized F includes the Essential Prime Implicants. Thus, we must eliminate the minterms that are part of an Essential Prime Implicant out of the other Prime Implicants: cross out the rows of the Essential Prime Implicants and the columns of the covered minterms. In the example, the Essential Prime Implicants are: $\overline{B}\overline{C}, C\overline{D}$
   - ✓ For the remaining X's: select enough Prime Implicants to cover all the minterms of the function. This is a trial and error procedure: start by selecting the Prime Implicant that crosses out (rows and columns) most of the Xs, and so on.

Prime Implicants		Minterms									
		0	1	2	5	6	7	8	9	10	14
$m_{0,1,8,9}$	$\bar{B}\bar{C}$	X	X					X	**X**		
$m_{0,2,8,10}$	$\bar{B}\bar{D}$	X		X				X		X	
$m_{2,6,10,14}$	$C\bar{D}$			X		X				X	**X**
$m_{1,5}$	$\bar{A}\bar{C}D$		X		X						
$m_{5,7}$	$\bar{A}BD$				**X**		**X**				
$m_{6,7}$	$\bar{A}BC$					X	X				

$$\rightarrow F(A,B,C,D) = \bar{B}\bar{C} + C\bar{D} + \bar{A}BD$$

EXAMPLE: $F(A,B,C,D) = \sum m(4,8,10,11,12,15) + \sum d(9,14)$. Function with don't care terms.

- ✓ **Implicants Table**: To help simplifying the function, the don't care terms are included as minterms here. If a don't care term ends up being a Prime Implicant, we delete it (otherwise we are not taking advantage of the don't care terms).

Number of ones	4-literal implicants	3-literal implicants	2-literal implicants	1-literal implicants
0				We can't combine any further, so we stop here
1	m_4 = 0100 ✓ m_8 = 1000 ✓	**$m_{4,12}$ = -100** $m_{8,9}$ = 100- ✓ $m_{8,10}$ = 10-0 ✓ $m_{8,12}$ = 1-00 ✓	**$m_{8,9,10,11}$ = 10--** ~~$m_{8,10,9,11}$ = 10--~~ **$m_{8,10,12,14}$ = 1--0** ~~$m_{8,12,10,14}$ = 1--0~~	
2	m_9 = 1001 ✓ m_{10}= 1010 ✓ m_{12}= 1100 ✓	$m_{9,11}$ = 10-1 ✓ $m_{10,11}$ = 101- ✓ $m_{10,14}$ = 1-10 ✓ $m_{12,14}$ = 11-0 ✓	**$m_{10,11,14,15}$ = 1-1-** ~~$m_{10,14,11,15}$ = 1-1-~~	
3	m_{11}= 1011 ✓ m_{14}= 1110 ✓	$m_{11,15}$ = 1-11 ✓ $m_{14,15}$ = 111- ✓		
4	m_{15}= 1111 ✓			

$$F(A,B,C,D) = B\bar{C}\bar{D} + A\bar{B} + A\bar{D} + AC$$

- ✓ **Prime Implicant Chart**: <u>The don't care terms are NOT included here</u>. Only the minterms are included here, since we are trying to have as few X's as possible.

Prime Implicants		Minterms					
		4	8	10	11	12	15
$m_{4,12}$	$B\bar{C}\bar{D}$	**X**				X	
$m_{8,9,10,11}$	$A\bar{B}$		**X**	X	X		
$m_{8,10,12,14}$	$A\bar{D}$		**X**	X		X	
$m_{10,11,14,15}$	AC			X	X		**X**

  - More than one minimal solution exist, depending on the **x** (in the same pink column) that we use:

$$\rightarrow F(A,B,C,D) = B\bar{C}\bar{D} + AC + A\bar{B}$$

$$Or: F(A,B,C,D) = B\bar{C}\bar{D} + AC + A\bar{D}$$

EXAMPLE: $F(A,B,C) = \sum m(0,1,2,5,6,7)$

- ✓ **Implicants Table**:

Number of ones	3-literal implicants	2-literal implicants	1-literal implicants
0	m_0 = 000 ✓	**$m_{0,1}$ = 00-** **$m_{0,2}$ = 0-0**	We can't combine any further, so we stop here
1	m_1 = 001 ✓ m_2 = 010 ✓	**$m_{1,5}$ = -01** **$m_{2,6}$ = -10**	
2	m_5 = 101 ✓ m_6 = 110 ✓	**$m_{5,7}$ = 1-1** **$m_{6,7}$ = 11-**	
3	m_7 = 111 ✓		

$$F(A,B,C) = \bar{A}\bar{B} + \bar{A}\bar{C} + \bar{B}C + B\bar{C} + AC + AB$$

✓ **Prime Implicant Chart**:

Prime Implicants		Minterms					
		0	1	2	5	6	7
$m_{0,1}$	$\bar{A}\bar{B}$	**X**	X				
$m_{0,2}$	$\bar{A}\bar{C}$	X		X			
$m_{1,5}$	$\bar{B}C$		X		X		
$m_{2,6}$	$B\bar{C}$			**X**		X	
$m_{5,7}$	AC				**X**		X
$m_{6,7}$	AB					X	X

- No essential prime implicants. So, we can only select the minimum number of Prime Implicants (i.e., crossing out rows and columns) that covers all the minterms. The example above shows a group of Prime Implicants whose number of elements is the minimum (3).
- For this particular group, there is only one minimal solution (recall that there can be more than one minimal solution for the same group of Prime Implicants):

$$F(A,B,C) = \bar{A}\bar{B} + B\bar{C} + AC$$

- There can be more than one group of Prime Implicants whose number of elements is the minimum (3 in this example), as we can pick any **x** in a column to mark our Prime Implicants. The example below shows the only other possible minimum group of Prime Implicants:

Prime Implicants		Minterms					
		0	1	2	5	6	7
$m_{0,1}$	$\bar{A}\bar{B}$	X	X				
$m_{0,2}$	$\bar{A}\bar{C}$	**X**		X			
$m_{1,5}$	$\bar{B}C$		**X**		X		
$m_{2,6}$	$B\bar{C}$			X		X	
$m_{5,7}$	AC				X		X
$m_{6,7}$	AB					**X**	X

- For this particular group of Prime Implicants, there is only one minimal solution:

$$F(A,B,C) = \bar{A}\bar{C} + \bar{B}C + AB$$

EXAMPLE: $F(A,B,C,D) = \sum m(0,2,3,5,6,7,8,9) + \sum d(10,11,12,13,14,15)$.

There are too many minterms. This will make the process cumbersome. Instead, it might be better to optimize:

$$\bar{F}(A,B,C,D) = \sum m(1,4) + \sum d(10,11,12,13,14,15)$$

Once we get the optimized form of $\bar{F}$, we complement it in order to get F.

✓ **Implicants Table ($\bar{F}$)** :

Number of ones	4-literal implicants	3-literal implicants	2-literal implicants	1-literal implicants
0				
1	**m_1 = 0001** m_4 = 0100 ✓	**$m_{4,12}$ = -100**		We can't combine any further, so we stop here
2	m_{10}= 1010 ✓ m_{12}= 1100 ✓	$m_{10,11}$ = 101- ✓ $m_{10,14}$ = 1-10 ✓ $m_{12,13}$ = 110- ✓ $m_{12,14}$ = 11-0 ✓	**$m_{10,11,14,15}$ = 1-1-** ~~$m_{10,14,11,15}$ = 1-1-~~ **$m_{12,13,14,15}$ = 11--** ~~$m_{12,14,13,15}$ = 11--~~	
3	m_{11}= 1011 ✓ m_{13}= 1101 ✓ m_{14}= 1110 ✓	$m_{11,15}$ = 1-11 ✓ $m_{13,15}$ = 11-1 ✓ $m_{14,15}$ = 111- ✓		
4	m_{15}= 1111 ✓			

$$\bar{F}(A,B,C,D) = \bar{A}\,\bar{B}\,\bar{C}D + B\bar{C}\bar{D} + AC + AB$$

- ✓ **Prime Implicant Chart ($\bar{F}$)**: The don't care terms are NOT included.

Prime Implicants		Minterms	
		1	4
m_1	$\bar{A}\,\bar{B}\,\bar{C}D$	X	
$m_{4,12}$	$B\bar{C}\bar{D}$		X
$m_{10,11,14,15}$	AC		
$m_{12,13,14,15}$	AB		

$$\rightarrow \bar{F}(A,B,C,D) = \bar{A}\,\bar{B}\,\bar{C}D + B\bar{C}\bar{D} \Rightarrow F(A,B,C,D) = (A+B+C+\bar{D})(\bar{B}+C+D)$$

- ✓ Note: This method obtains the most minimized form of $\bar{F}$ as a SOP: $\bar{F} = \bar{A}\,\bar{B}\,\bar{C}D + B\bar{C}\bar{D}$. For F, it obtains the most minimized form of F as a POS: $F = (A+B+C+\bar{D})(\bar{B}+C+D)$.

- ✓ We can always get F as a SOP, but note that the expression needs some minimization:
$F = A\bar{B} + AD + BD + \bar{D}\bar{B} + C + CD + CA + CB + C\bar{B} + C\bar{D} = DA + \bar{D}\bar{B} + A\bar{B} + BD + C = DA + \bar{D}\bar{B} + BD + C$ (*)

- ✓ If we apply Quine-McCluskey directly to F, the resulting Boolean SOP expression might look different than $F = DA + \overline{DB} + BD + C$. In fact, it might be even more minimized (note how we needed to apply some minimization to F in *). This is because the don't care terms are likely to be assigned '1' and '0' differently for F and $\bar{F}$. The resulting functions will be equivalent though.

- ✓ Conclusion: For any function, using the method of minimizing $\bar{F}$ and then invert to get F, the resulting expression of F as a SOP might appear different (and be not fully minimized) than the F resulting by applying Quine-McCluskey directly to F.

ISSUES:

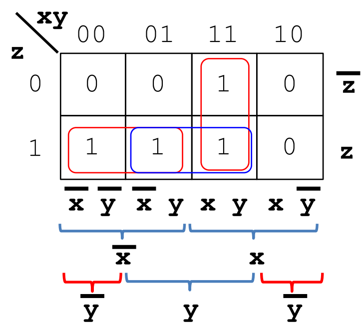
- To determine a minimal solution (i.e. solution with the same number of literals), we need to efficiently cross out rows and columns. We can do this by trial and error, but it can become a cumbersome procedure as the number of variables increase. And as illustrated in the examples, there can be more than one way to efficiently cross out rows and columns.

- There can also be more than one minimal solution (even if there is only one way to efficiently cross out rows and columns) resulting from this method. We can determine all possible minimal solutions by inspection, but this can become cumbersome as the number of variables increase.

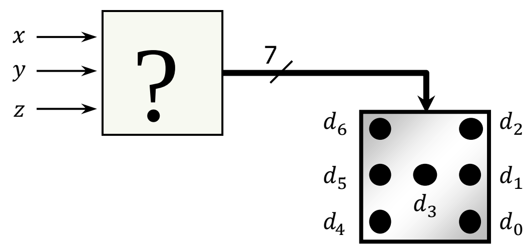
- A systematic way to determine all possible minimum solutions is provided by **Petrick's method**: given a prime implicant chart, we can determine all minimum sum-of-products solutions. This is out of the scope of this course.

PRACTICE EXERCISES

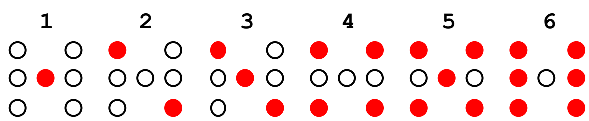
- The figure depicts the K-map of a 3-variable Boolean function.
  - ✓ Provide the simplified Boolean equation.
  - ✓ Is the term resultant from the blue box redundant? Why?
  - ✓ What Boolean Theorem is this K-map illustrating?



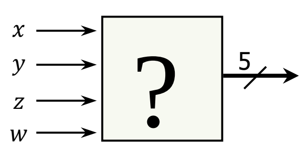
- We want to display the results of a roll of a die on a 7-LED array. The input data is a 3-bit unsigned number from 1 to 6. For example, the code 101 needs to be displayed such that it represents the number '5' in the die-like LED array. For the given arrangement of Boolean variables in the LED array (a, b, c, d, e, f, g), the number '5' is represented as '1011101'.
  - ✓ Design the logic circuit that converts the 3-bit value to a 7-bit LED pattern (LED ON represents a logic '1', and LED OFF represents a logic '0'): complete the truth table for each output $abcdefg$, provide the simplified expressions, and sketch the logic circuit.



x y z	Die value
0 0 1	**1**
0 1 0	**2**
0 1 1	**3**
1 0 0	**4**
1 0 1	**5**
1 1 0	**6**



- Design a circuit that converts a 4-bit unsigned integer (whose values range from 0 to 9) into Morse code (alphanumeric characters are encoded into sequences of dots and dashes). The figure depicts the Morse code representations for numbers from 0 to 9. The circuit generates 5 bits, where a dot is represented as a '0', and a dash is represented as a '1'.



Decimal value	Morse code
0	▬ ▬ ▬ ▬ ▬
1	• ▬ ▬ ▬ ▬
2	• • ▬ ▬ ▬
3	• • • ▬ ▬
4	• • • • ▬
5	• • • • •
6	▬ • • • •
7	▬ ▬ • • •
8	▬ ▬ ▬ • •
9	▬ ▬ ▬ ▬ •

- Simplify the Boolean expression for f. Note that an 'X' on the input means that the logical value can be either '0' or '1'. So, if the input $xyzw$ is 0X1X, it means that for the output f to be 1, we only need $x = 0$, and $z = 1$.

x y z w	f
0 X 1 X	1
X 1 0 1	1
1 1 X 0	1
1 0 0 1	1
all others	0