CMSC 130 – Logic Design and Digital Computer Circuits
Handout # 4: SIMPLIFICATION OF LOGIC CIRCUITS

3 methods used in Simplification

1. Boolean Algebra.
2. Karnaugh Map.
3. Quine-McCluskey.

BOOLEAN ALGEBRA

Simplification using this method is done by using the theorems of Boolean Algebra to simplify logic circuits.

K-MAP

This method makes use of the graphical representation of the boolean function called the K-MAP. The K-MAP consists cells wherein each cell corresponds to a minterm entry with the whole table wrapping around vertically and horizontally. Various patterns are used in order to simplify the boolean function.

K-MAP patterns:

- Singles.
- Pairs.
- Quads.
- Octets.

2-var K-MAP

X \ Y	0	1
0	m_0	m_1
1	m_2	m_3

3-var K-MAP

X \ YZ	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

4-var K-MAP

WX \ YZ	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

5-var K-MAP

VW \ XYZ	000	001	011	010	110	111	101	100
00	m_0	m_1	m_3	m_2	m_6	m_7	m_5	m_4
01	m_8	m_9	m_{11}	m_{10}	m_{14}	m_{15}	m_{13}	m_{12}
11	m_{24}	m_{25}	m_{27}	m_{26}	m_{30}	m_{31}	m_{29}	m_{28}
10	m_{16}	m_{17}	m_{19}	m_{18}	m_{22}	m_{23}	m_{21}	m_{20}

Example: Simplify the function f = ∑(0, 2, 3, 4, 5, 8, 12, 13)

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

*minimize the number of groups inside the K-MAP.

f = c'd' + bc'+ a'b'c

DON'T CARES – Operations that never occur during normal operation. These are marked as X in the truth table and is treated as a 1 or a 0 to your advantage.

QUINE-MCCLUSKEY METHOD

I. Determine the prime implicants.

a. Form sections by grouping binary representation of the minterms according to the number of 1's contained.
b. Any two minterms that differ from each other by one and only one variable can be combined and the unmatched variable removed. Compare the minterms in one section with those of the next section. If any two minterms are the same in every position but one, check it to mark that it has been used. The resulting terms are placed in column b together with their decimal equivalents. The variable eliminated is indicated by a dash.
c. The searching and comparing process is repeated in subsequent columns as long as proper matching is encountered.
d. The unchecked terms in the table from the prime implicants.

II. Select the prime implicants.

a. Place all prime implicants in a column and each minterm in rows to form a matrix. Place crosses in each row to indicate the minterms contained in the prime implicant of that row as indicated by the decimal equivalent of the term.
b. Place a check under the column of the minterm with only 1 x. place a check in the side of the prime implicants that covers these minterms to indicate that they have been selected. These prime implicants are called *essential prime implicants*.
c. Place a check on each column whose minterm is covered by the selected prime implicants in b.
d. Examine those columns without any check. The prime implicant that covers these unchecked columns is selected.

III. The sum of the selected prime implicants form the simplified version of the function.

Example: Simplify the function f = ∑ (1,4,6,7,8,9,10,11,15)

(1)

	a	b	c	d	
1	0	0	0	1	✓
4	0	1	0	0	✓
8	1	0	0	0	✓
6	0	1	1	0	✓
9	1	0	0	1	✓
10	1	0	1	0	✓
7	0	1	1	1	✓
11	1	0	1	1	✓
15	1	1	1	1	✓

(2)

	a	b	c	d	
1,9	--	0	0	1	
4,6	0	1	—	0	
8,9	1	0	0	—	✓
6,7	0	1	1	—	
9,11	1	0	—	1	✓
10,11	1	0	1	—	✓
7,15	--	1	1	1	
11,15	1	--	1	1	

(3)

	a	b	c	d
8,9,10,11	1	0	—	--

Prime Implicants: b'c'd, a'bd', a'bc, bcd, acd, and ab'

Selection of prime implicants:

		1	4	6	7	8	9	10	11	15
✓b'c'd	1,9	X					X			
✓a'bd'	4,6		X	X						
a'bc	6,7			X	X					
bcd	7,15				X					X
acd	11,15								X	X
✓ab'	8,9,10,11					X	X	X	X	
		✓	✓	✓		✓	✓	✓	✓	

Without check: 7 and 15. This is covered by 7,15 or the implicant bcd.
Get the checked implicants plus bcd.

Simplified function: f= b'c'd + a'bd' + ab' + bcd