ECE380 Digital Logic

Optimized Implementation of Logic Functions:

Karnaugh Maps and Minimum Sum-of-Product Forms

Karnaugh map

- The key to finding a minimum cost SOP or POS form is applying the combining property (14a for SOP or 14b for POS)
- The ***Karnaugh map*** (K-map) provides a systematic (and graphical) way of performing this operation
- Minterms can be combined by 14a when they differ in only one variable
  - $f(x,y,z) = xyz+xyz' = xy(z+z') = xy(1) = xy$
- The K-map illustrates this combination graphically

Karnaugh map

- The K-map is an alternative to a truth table for representing an expression
  - K-map consists of cells that correspond to rows of the truth table
  - Each cell corresponds to a minterm
- A two variable truth table and the corresponding K-map

x_1	x_2	f
0	0	m_0
0	1	m_1
1	0	m_2
1	1	m_3

$x_2 \backslash x_1$	0	1
0	m_0	m_2
1	m_1	m_3

Karnaugh map

Values for the first variable are listed across the top

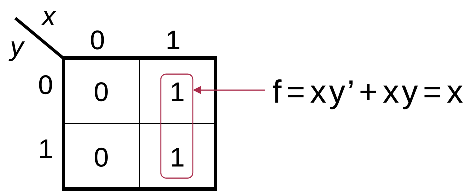
$x_2 \backslash x_1$	0	1
0	m_0	m_2
1	m_1	m_3

Values for the second variable are listed down the left side

Karnaugh map groupings

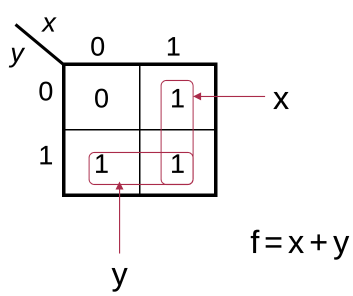
- Minterms in adjacent squares on the map can be combined since they differ in only one variable
- Indicated by looping the corresponding ‘1’s on the map (the ‘1’s must be adjacent)
- Looping two ‘1’s together corresponds to eliminating a term and a variable from the output expression => xy+xy’ = x

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



K-map groupings example

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



- Note that the bottom two cells differ in only one variable (x) and the right two cells differ in only one variable (y)

K-map groupings example

- Draw the K-map and give the minimized logic expression for the following truth table.
- Show the groupings made in the K-map

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

Electrical & Computer Engineering Dr. D. J. Jackson Lecture 7-7

Three variable K-map

- A three-variable K-map is constructing by laying 2 two-variable maps side by side
- K-map are always laid out such that adjacent squares only differ by one variable (i.e. by 1 bit in the binary expression of the minterm values)

x	*y*	*z*	*Minterm*
0	0	0	$m_0=x'y'z'$
0	0	1	$m_1=x'y'z$
0	1	0	$m_2=x'yz'$
0	1	1	$m_3=x'yz$
1	0	0	$m_4=xy'z'$
1	0	1	$m_5=xy'z$
1	1	0	$m_6=xyz'$
1	1	1	$m_7=xyz$

z \ *xy*	00	01	11	10
0	m_0	m_2	m_6	m_4
1	m_1	m_3	m_7	m_5

End cells are 'adjacent'

Electrical & Computer Engineering Dr. D. J. Jackson Lecture 7-8

Example three-variable K-maps

$f(x,y,z)=\Sigma m(0,1,2,4)$
$=x'y'+x'z'+y'z'$

z \ xy	00	01	11	10
0	1	1	0	1
1	1	0	0	0

$f(x,y,z)=\Sigma m(0,1,2,3,4)$
$=x'+y'z'$

z \ xy	00	01	11	10
0	1	1	0	1
1	1	1	0	0

A grouping of four eliminates 2 variables

Guidelines for combining terms

- Can combine only adjacent '1's
- Can group only in powers of 2 (1,2,4,8, etc.)
- Try to form as large a grouping as possible
- Do not generate more groups than are necessary to "cover" all the '1's

Example groupings

z \ xy	00	01	11	10
0	1	1	1	1
1	0	0	0	0

f=z'

z \ xy	00	01	11	10
0	0	1	1	1
1	0	0	1	1

f=yz'+x

z \ xy	00	01	11	10
0	1	1	1	1
1	1	0	0	1

f=z'+y'

z \ xy	00	01	11	10
0	1	1	1	0
1	0	1	1	0

f=y+x'z'

K-map groupings example

- Draw the K-map and give the minimized logic expression for the following.
  - f(a,b,c)=Σm(1,2,3,4,5,6)
- Show the groupings made in the K-map

Four variable K-map

- A four-variable K-map is constructing by laying 2 three-variable maps together to create four rows
  - f(a,b,c,d)

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

Four variable K-map

- Adjacencies wrap around in the K-map

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

Example four-variable K-maps

$f(a,b,c,d) = \Sigma m(2,3,9\text{-}11,13)$

$= ac'd + b'c$

cd \ ab	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	1	0	0	1
10	1	0	0	1

$f(a,b,c,d) = \Sigma m(3\text{-}7,9,11,12\text{-}15)$

$= b + cd + ad$

cd \ ab	00	01	11	10
00	0	1	1	0
01	0	1	1	1
11	1	1	1	1
10	0	1	1	0

Example groupings

cd \ ab	00	01	11	10
00	1	1	1	1
01	1	0	0	1
11	1	0	0	1
10	1	1	1	1

$f(a,b,c,d) = b' + d'$

cd \ ab	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	0	0	1
10	0	1	1	0

$f(a,b,c,d) = b'd + bd'$

Example groupings

cd \ ab	00	01	11	10
00	1	0	0	1
01	0	1	1	0
11	0	1	1	0
10	1	0	0	1

$f(a,b,c,d) = b'd' + bd$

cd \ ab	00	01	11	10
00	1	1	1	0
01	1	0	0	1
11	1	0	0	1
10	1	1	1	0

$f(a,b,c,d) = b'd + bd' + a'b'$