EE 2720

Handout # 14

<u>Question:</u> Do you have any questions so far?
<u>Answer:</u> Yes; No; I don't know; I don't care;
I am not attending class today (too bad!!) so I
can't answer the question. Well, I couldn't
think of any other possible answers.

Well, I have a question.

<u>Question:</u> Are you tired of algebra? If so, I
have some good news for you. We already saw
that the topic of simplifying a switching fun-
ction is very important; (it can significantly
reduce the cost of a circuit realization). You can
do the above task by using algebraic techni-
ques (theorems of switching algebra) but you
can also do it <u>with less effort using gra-
phical techniques.</u> Again, you will enjoy the
graphical technique but you have to wait
a little bit. Let me get started.

• Switching functions can generally be simplified
by using algebraic techniques (theorems of
switching algebra). However, two problems arise
when algebraic procedures are used:

1. The procedures are difficult to apply in a sys-
tematic way.

2. It is difficult to tell when you have arrived
at a minimum solution.

The Karnaugh map method (this is the ② graphical technique or approach that I told you above) which will be studied in this handout and probably future handouts overcomes the above mentioned difficulties by providing a systematic method for simplifying switching functions. The Karnaugh map is an especially useful tool for simplifying and manipulating switching functions of three or four variables, but it can be extended to functions of five or more variables. Generally, you will find that the Karnaugh map method is faster and easier to apply than other simplification methods.

Let me demonstrate the simplification of switching functions by using algebraic techniques for the last time (at least last time for this handout; why would we do it again since we are going to learn a more convenient technique which is the Karnaugh map technique?). I will demonstrate the above with two examples.

• <u>Simplifying a switching function given in canonical sum form:</u> You already know that any switching function can be expressed as a canonical sum. This is a non-simplified sum-of-products form. To simplify the canonical sum:

1. Combine terms by using the theorem

given product term · Y + given product term · Y' = given product term ①.

Note: The above eq. (1) is a generalization of theorem (T10) and was stated and proved in handout # 13. A given term may be used more than once because $X + X = X$.

2. Eliminate redundant terms by using the consensus theorem (T11) or other theorems.

Note: Unlike the canonical sum which is unique, the obtained simplified sum-of-products is not necessarily unique. That is, for a given function, you might have two simplified sum-of-products, each with the same number of terms and the same number of literals.

• Simplifying a switching function given in canonical product form: You already know that any switching function can be expressed as a canonical product. This is a non-simplified product-of-sums form. To simplify the canonical product:

1. Combine terms by using the theorem:

$$(given \; sum \; term + Y) \cdot (given \; sum \; term + Y') = given \; sum \; term \quad (2)$$

Note: The above eq. (2) is a generalization of theorem (T10') and was stated and proved in handout # 13.

A given term may be used more than once because $X \cdot X = X$

2. Eliminate redundant terms by using the consensus theorem (T11') or other theorems.

Note: Unlike the canonical product which is
unique, the obtained simplified product-of-sums is
not necessarily unique.

Example 1: Simplify the function F where F is

$$F = \Sigma_{a,b,c}(0,1,2,5,6,7).$$

Answer: We have:

$$F = \Sigma_{a,b,c}(0,1,2,5,6,7) =$$
$$= a' \cdot b' \cdot c' + a' \cdot b' \cdot c + a' \cdot b \cdot c' + a \cdot b' \cdot c + a \cdot b \cdot c' + a \cdot b \cdot c$$

combine   combine   combine   combine

$$= a' \cdot b' + b' \cdot c + b \cdot c' + a \cdot b ; (\text{In the above I silently applied } X+X=X).$$

None of the terms in the above expression can be
eliminated by the consensus theorem. However, combi-
ning terms in a different way leads to a more
simplified sum-of-products (the minimal one).

$$F = a' \cdot b' \cdot c' + a' \cdot b' \cdot c + a' \cdot b \cdot c' + a \cdot b' \cdot c + a \cdot b \cdot c' + a \cdot b \cdot c$$

combine   combine   combine

$$= a' \cdot b' + b \cdot c' + a \cdot c$$

Example 2: Simplify the function F where F is

$$F = \Pi_{A,B,C,D}(2,5,6,7,10,14).$$

Answer: We have:

$$F = \Pi_{A,B,C,D}(2,5,6,7,10,14) =$$

$$= \Pi_{A,B,C,D} (5,7,6,14,2,10) =$$
$$(A+B'+C+D') \cdot (A+B'+C'+D') \cdot (A+B'+C'+D) \cdot (A'+B'+C'+D) \cdot$$

combine   combine   combine

$$\cdot (A+B+C'+D) \cdot (A'+B+C'+D) =$$

combine

$$= (A+B'+D') \cdot (A+B'+C') \cdot (B'+C'+D) \cdot (B+C'+D)$$

combine

$$= (A+B'+D') \cdot (A+B'+C') \cdot (C'+D) =$$

eliminate by consensus theorem (T11')

$$= (A+B'+D') \cdot (C'+D); \text{ (this is the most simplified}$$
product-of-sums expression).

Note: In the above, I silently applied $X \cdot X = X$

• Karnaugh Maps

A Karnaugh map is a graphical representation of the truth table of a logic function. The figures 1, 2, 3 so on next page show Karnaugh maps for logic functions of 2, 3 and 4 variables respectively.
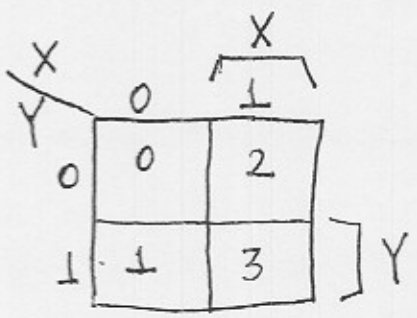
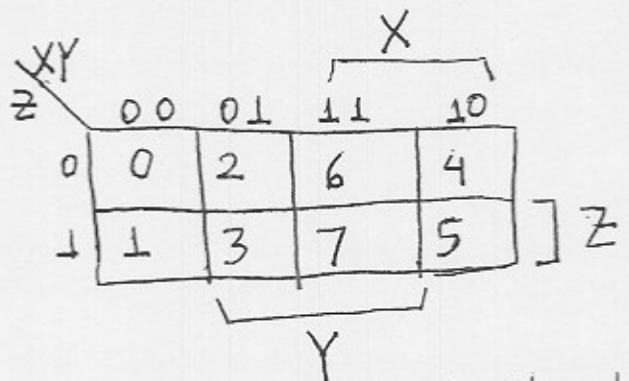**Figure 1: Karnaugh map for logic function of 2 variables.**

	X=0	X=1
Y=0	0	2
Y=1	1	3

Figure 2: 3-variable Karnaugh map.

XY across (00, 01, 11, 10), Z down (0, 1):

Z \ XY	00	01	11	10
0	0	2	6	4
1	1	3	7	5

Figure 3: 4-variable Karnaugh map.

WX across (00, 01, 11, 10), YZ down (00, 01, 11, 10):

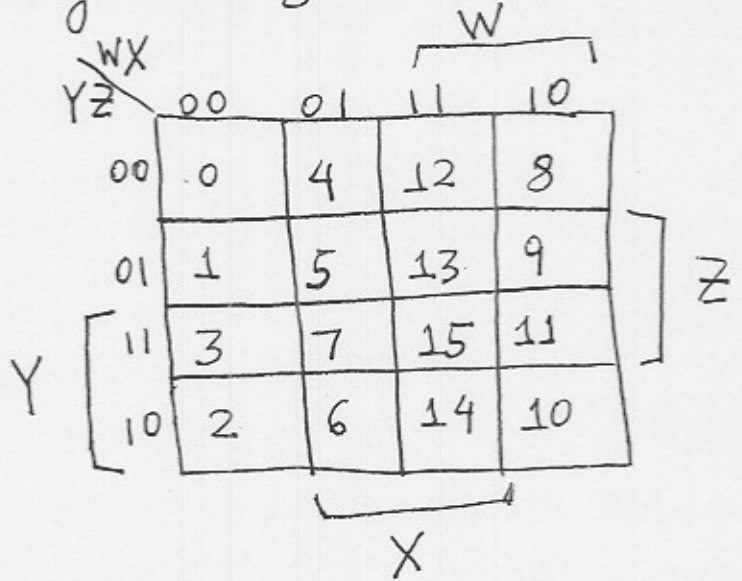| YZ \ WX	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

The Karnaugh map for an n-input logic function is an array with 2^n cells, one for each possible input combination or minterm. The rows and columns of a Karnaugh map are labeled. The numbers inside each cell are the corresponding minterm num

bers in the truth table. Truth-table inputs are labeled alphabetically from left to right (for example X, Y, Z) and the rows are numbered in binary counting order. For example, cell 12 in the 4-variable Karnaugh map of fig. 3 corresponds to the truth-table row in which $WXYZ = 1100$, cell 5 in the 4-variable map of fig. 3 corresponds to the truth-table row in which $WXYZ = 0101$ etc. For a given function, each cell of the karnaugh map of this function contains the information from the respective raw of the truth table of the function; (a 0 if the function is 0 for that input combination; a 1 otherwise).

• <u>Brackets</u>: As seen from figures 1, 2, 3 we used brackets; (⌐ or [ ). Each bracketed region is the part of the map in which the indicated variable is 1; (brackets provide same information given by row and column labels).

• To represent a logic function on a Karnaugh map, we copy 0s and 1s from the truth table to the corresponding cells of the map.

• <u>Ordering of row and column numbers</u>: Each cell corresponds to an input combination that differs from each of its immediately adjacent neighbors in only

one variable. For example, cells 5 and 7 in the 4-variable map of fig. 3 differ only in the value of Y, cells 7 and 15 in the same map differ only in the value of W etc. <u>Be careful</u>. In the 3- and 4-variable maps of figures 2 and 3, cells on the left/right or top/bottom borders are still adjacent. For example, cells 2 and 10 in the 4-variable map of fig. 3 are adjacent because they differ only in the value of W, cells 4 and 6 in the same map are adjacent because they differ only in the value of Y etc; (this is called cell wraparound).

~~Each cell is associated with a corresponding minterm~~
~~table corresponds to a minterm~~

- <u>Minimizing Sums of Products</u>

  For a given logic function, each input combination with a 1 in the truth table corresponds to a min-term in the canonical sum of the logic function. Since pairs of adjacent 1 cells in the Karnaugh map have minterms that differ only in one va-riable, the minterm pairs can be combined into a single product term by using the theorem of eq. (1) which was stated at the bottom of page 2. <u>Thus, a Karnaugh map can be used to sim-plify the canonical sum of a logic function.</u>

- In general, we can simplify a logic function using

a Karnaugh map by combining pairs of adj-
acent 1-cells (minterms) whenever possible and wri-
ting a sum of product term that covers all of the
1-cells. ~~The number of adjacent cells~~

- Many times, we can combine more than two ~~cells~~
  ~~~~ 1-cells into a single product term; (algebra-
ically this is based again on applying theorem
of eq. (1) of page 2 many times (iteratively):
I'll show you an example later.).

- The number of adjacent 1-cells being combi-
ned must be a power of 2.

- In general, 2^i 1-cells can be combined to form
a product term containing $\underline{n-i}$ literals, where
$\underline{n}$ is the number of variables in the function;
(that means that $\underline{i}$ literals are eliminated;
recall that each original term contains $\underline{n}$ li-
terals; reminder: literal is a variable or the
complement of a variable)

- A set of 2^i 1-cells may be combined if there
are $\underline{i}$ variables of the logic function that ta-
ke on all 2^i possible combinations within
that set, while the remaining $\underline{n-i}$ variables
have the same value throughout that set. The
corresponding product term has $\underline{n-i}$ literals,
where a variable is complemented if it appears
as 0 in all of the 1-cells, and uncomplemented
if it appears as 1.

Let me explain this last statement with an exam-
ple on next page.

Consider the 3-variable Karnaugh map of fig. 2 . Suppose we want to combine cells 0,1,4,5 and suppose these cells contain ones. Two figures are shown below:

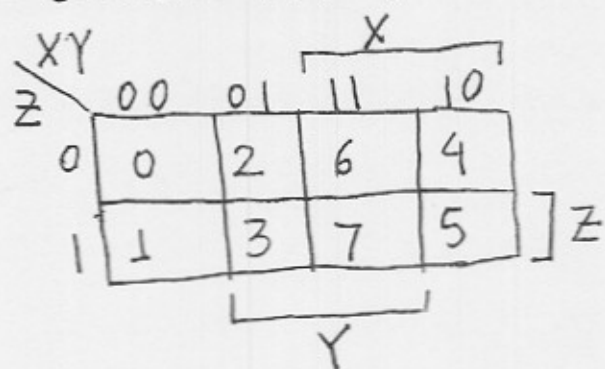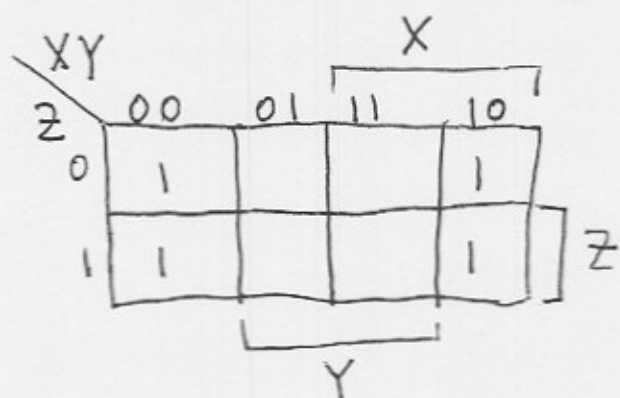Figure 4: 3-variable Karnaugh map.

Figure 5: Here I show that cells 0,1,4,5 contain ones.

Here the number of cells to be combined is $4 = 2^2$ so $i = 2$. The number of variables is $n = 3$. Here there are $i = 2$ variables (X and Z) that take on all $2^i = 2^2 = 4$ possible combinations within the set of cells 0,1,4,5 and the remaining $n - i = 3 - 2 = 1$ variable which is Y has the same value (which is 0) throughout the set of cells 0,1,4,5.

The corresponding product term will have $n - i = 3 - 2 = 1$ literal where the variable Y will be complemented since it appears as 0 in all the 1-cells 0, 1, 4, 5. So we will get Y' as a result of combining cells 0, 1, 4, 5.

<u>Note</u>: Examples on Karnaugh maps will be shown in the next handout. The subject of Karnaugh maps will be continued in the next handout. This was just an introduction.