EE 2720

Handout #15.

• Karnaugh maps continued.

We said in handout# 14 that the main mathematical basis for minimizing sum of products using a Karnaugh map is the theorem of equation (1) provided below:

given product term • Y + given product term • Y' = given product term (1).

Note: The above theorem of eq. (1) is a generalization of theorem (T10) and was stated and proved in handout# 13. It was also stated in handout# 14. Some of the following examples will clarify the above.

Example 1: Find a simplified sum-of-products expression for the logic function $F = \sum_{X,Y,Z}(1,2,5,7)$.

Answer: I will provide a truth table and a Karnaugh map for the logic function F. These are shown in tables 1 and 2 below respectively.

| X Y Z | F |
|---|---|
| 0 0 0 | 0 |
| 0 0 1 | 1 |
| 0 1 0 | 1 |
| 0 1 1 | 0 |
| 1 0 0 | 0 |
| 1 0 1 | 1 |
| 1 1 0 | 0 |
| 1 1 1 | 1 |

Table 1: Truth table for the function F of the example 1.

The Karnaugh map is shown on next page.

| Z \ XY | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | 1 | | |
| 1 | 1 | | 1 | 1 |

X: columns 11, 10; Y: columns 01, 11; Z: row 1

Table 2: Karnaugh map for the function F of example 1. Here I show only the ones, not the zeros, but anyways we are going to combine cells containing ones.

<u>Note</u>: It would have been better if I had called table 2 figure 1 but I hope you don't mind!

Below I show a karnaugh map that shows the combining of adjacent 1-cells. This is shown in figure 1 below:

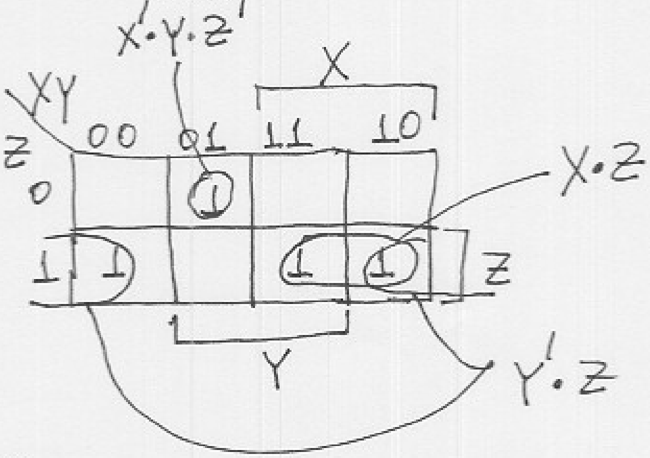


Figure 1: Combining adjacent 1-cells for the function F of example 1.

Let me explain algebraically what happened above in figure 1 (combining adjacent 1-cells I mean). I will use theorem of eq. (1) stated on page 1.

→Go to next page →

Consider cells 5 and 7 in fig. 1 of the previous page. Their contribution to the canonical sum for function F is:

$$X \cdot Y' \cdot Z + X \cdot Y \cdot Z = X \cdot Z \cdot Y + X \cdot Z \cdot Y' = X \cdot Z$$

Cousider now cells 1 and 5. Due to cell wraparound, these cells are adjacent. The contribution of these cells to the canonical sum for function F is:
$X' \cdot Y' \cdot Z + X \cdot Y' \cdot Z = Y' \cdot Z$.

Note that cell 2 cannot be combined with any other cell because it is not adjacent to any other cell. So the overal result of simplification (simplified sum-of-products expression) for F is:

$$F = X' \cdot Y \cdot Z' + X \cdot Z + Y' \cdot Z \qquad (2)$$

Note: In the above, I silently applied the theorem $X + X = X$. Let me show you the entire picture algebraically.

$$F = \Sigma_{X,Y,Z}(1,2,5,7) = X' \cdot Y' \cdot Z + X' \cdot Y \cdot Z' + X \cdot Y' \cdot Z +$$
$$+ X \cdot Y \cdot Z = X' \cdot Y \cdot Z' + X' \cdot Y' \cdot Z + X \cdot Y' \cdot Z + X \cdot Y' \cdot Z +$$
$$+ X \cdot Y \cdot Z =$$
$$= X' \cdot Y \cdot Z' + (X' \cdot Y' \cdot Z + X \cdot Y' \cdot Z) + (Y' \cdot X \cdot Z + Y \cdot X \cdot Z) =$$
$$= X' \cdot Y \cdot Z' + Y' \cdot Z + X \cdot Z.$$ This is the same as eq. (2) above. As seen from the above, I duplicated term $X \cdot Y' \cdot Z$. But that is what happened in the Karnaugh map of fig 1. We

Combined cell 5 once with cell 7 and once with cell 1 and the contribution of cell 5 is $X \cdot Y' \cdot Z$ which is the term that I duplicated.

<u>Note</u>: Did you get the graphical approach? Are you ready to give up algebra? If you didn't get it here I am to explain.

- Cell 2 can't be combined with any other cell as we already explained; (remember why?). Its contribution to the canonical sum is $X' \cdot Y \cdot Z'$.
- Cells 5 and 7 are adjacent so they can be combined. Within that ~~cell~~ set of cells, variable Y takes on all possible values (0 and 1 I mean), while variables X and Z have the same value throughout the set of cells 5 and 7. The resulting product term as a result of combining cells 5 and 7 will have 2 literals, where both variables X and Z (these two variables will participate in the product term) will be <u>uncomplemented</u> because both X and Z appear as 1s in both cells 5 and 7. Therefore, the resulting product term (result of combining cells 5 and 7 I mean) will be $X \cdot Z$. Is it more clear now??
- Cells 1 and 5 are adjacent <u>due to wrap-around</u> so they can be combined. Within that set of cells, variable X takes on all possible values (0 and 1 I mean), while variables Y and Z have the same value throughout the set of cells 1 and 5. The resulting product term (as a result

of combining cells 1 and 5) will have 2 literals. The participating variables in the product term will be Y and Z. Variable Y will be <u>complemented</u> because it appears as 0 in both cells 1 and 5 while variable Z will be <u>uncomplemented</u> because it appears as 1 in both cells 1 and 5. Therefore, the resulting product term (result of combining cells 1 and 5) will be $Y' \cdot Z$. Is it VERY clear now??!!

To put the entire picture together now we have:

$F =$ (contribution of cell 2) $\overset{\leftarrow OR}{+}$ (result of combining cells 5 and 7) $\overset{\leftarrow OR}{+}$ (result of combining cells 1 and 5) $=$

$= X' \cdot Y \cdot Z' + X \cdot Z + Y' \cdot Z$. This equation is the same as eq. (2) on page 3 but I did not apply any algebra at all!! I only used the graphical approach named the Karnaugh map approach.

<u>Question:</u> Is this Karnaugh map approach clear now? If the answer is yes, then you are ready to forget about algebra. If the answer is no, then you can ask me. That is why you need teachers. Otherwise, you could study the book, submit HWs take exams and pass or ~~fail~~ not the course!!

Eq. (2) of page 3 suggests the simplified AND-OR logic circuit shown in fig. 2 on next page.

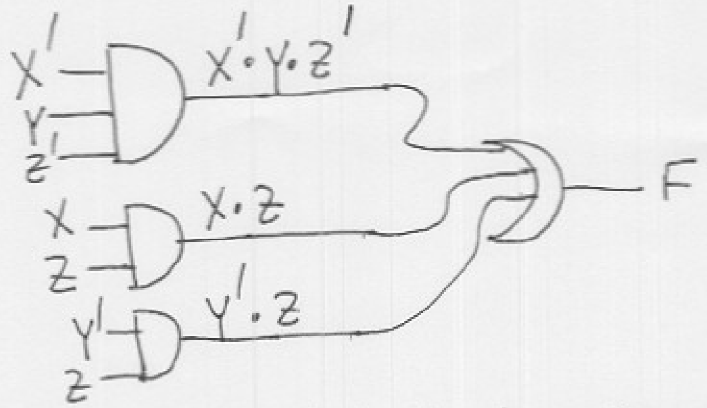


Figure 2: Minimized AND–OR circuit for the function F of example 1. Here I don't show the inverters. I hope it is OK with you!!

Note: By using the graphical approach and inserting bubbles, you can very easily transform the circuit of fig. 2 above into an equivalent NAND–NAND circuit. Do it. We have done such things a billion times, so by now it should be trivial!!

Note: In the previous example (example 1 I mean) we combined two cells of the Karnaugh map at a time. Sometimes, we can combine more than two 1-cells into a single product term. We can combine 4, 8, 16, etc... cells; (the number of cells to be combined must be a power of 2). Combining more than two cells into a single product term is again based algebraically on applying theorem of eq. (1) of page 1 many times (iteratively). The example that follows clarifies this issue.

Example 2: Find a simplified sum-of-products expression for the logic function $F = \Sigma_{X,Y,Z}(0, 1, 4, 5, 6)$.

Answer: I will first do it algebraically and then using a Karnaugh map. I believe this is the last time I use algebra (why would I use it again?). When I do it algebraically, I will use theorem of eq. (1) of page 1 several times (iteratively) and also theorem that states $X+X=X$. We have:

$F=\sum_{X,Y,Z}(0,1,4,5,6) = X'\cdot Y'\cdot Z' + X'\cdot Y'\cdot Z + X\cdot Y'\cdot Z' + X\cdot Y'\cdot Z + X\cdot Y\cdot Z' = X'\cdot Y'\cdot Z' + X'\cdot Y'\cdot Z + X\cdot Y'\cdot Z' + X\cdot Y'\cdot Z + X\cdot Y\cdot Z' + X\cdot Y'\cdot Z' = (X'\cdot Y'\cdot Z' + X'\cdot Y'\cdot Z) + (X\cdot Y'\cdot Z' + X\cdot Y'\cdot Z) + (X\cdot Y\cdot Z' + X\cdot Y'\cdot Z') = X'\cdot Y' + X\cdot Y' + X\cdot Z' = (X'\cdot Y' + X\cdot Y') + X\cdot Z' = Y' + X\cdot Z'$ (3).

I will now do the simplification of F using a Karnaugh map. The map is shown in figure 3 below:

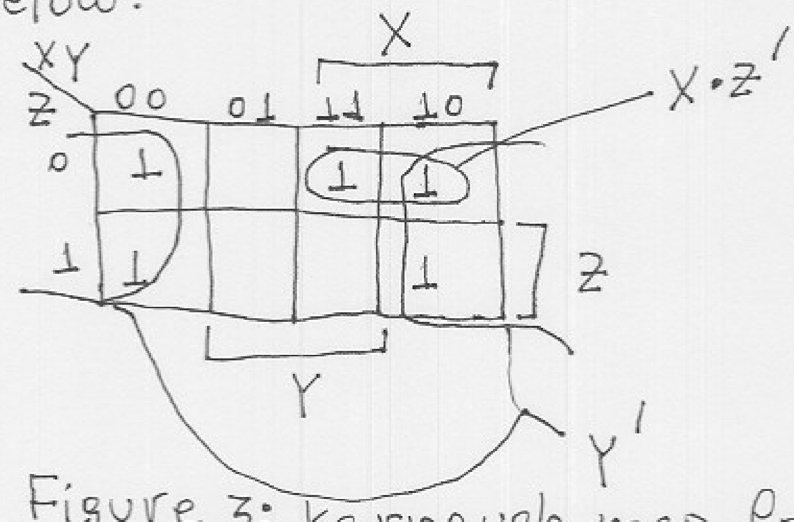


Figure 3: Karnaugh map for $F=\sum_{X,Y,Z}(0,1,4,5,6)$. The map shows the combining of cells.

I will now explain the above Karnaugh map. This is the last time I provide these detailed explanations.

- I am combining cells 4 and 6. Within this set of cells, variable Y takes on all possible values (0 and 1 I mean), while variables X and Z have the same value throughout the set of cells 4 and 6. The resulting product term, as a result of combining cells 4 and 6, will have 2 literals. The participating variables in the product term will be X and Z. Variable $\underline{X}$ will be <u>uncomplemented</u> because it appears as 1 in both cells 4 and 6 while variable $\underline{Z}$ will be <u>complemented</u> because it appears as 0 in both cells 4 and 6. Therefore, the resulting product term (result of combining cells 4 and 6 I mean) will be $X \cdot Z'$.

- I am combining cells 0, 1, 4, 5. Within this set of cells, variables X and Z take on all possible values (00, 01, 10, 11 I mean), while variable Y has the same value throughout the set of cells 0, 1, 4, 5. The resulting product term, as a result of combining cells 0, 1, 4, 5 will have 1 literal. The participating variable in the product term will be Y. Variable $\underline{Y}$ will be <u>complemented</u> because it appears as 0 in all cells 0, 1, 4, 5. Therefore, the resulting product term (result of combining cells 0, 1, 4, 5 I mean) will be $Y'$.
- We now covered (took into consideration) all 1's in the Karnaugh map, so we are ready to provide a simplified sum-of-products expression for the function F. We have:

$F = $(result of combining cells 4 and 6) $\overset{OR}{+}$ (result of combining cells 0, 1, 4, 5) $= X \cdot Z' + Y'$. (4).

<u>Question</u>: When I performed the algebraic de-

rivations on page 7, I duplicated the product term $X \cdot Y' \cdot Z'$ by theorem $X + X = X$. Did I do the same thing in the karnaugh map?

<u>Answer:</u> Yes I did. I combined cell 4 once with cell 6 and once with cells 0, 1, 5. And the contribution of cell 4 is $X \cdot Y' \cdot Z'$ which is the term that I duplicated.

Equation (4), (bottom of previous page), suggests the simplified AND-OR logic circuit for $F = \Sigma_{X,Y,Z}(0,1,4,5,6)$ shown in figure 4 below:

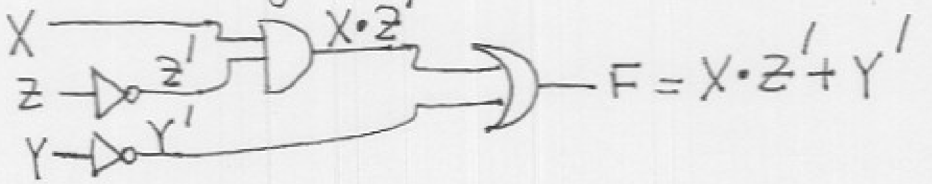


Figure 4: Simplified AND-OR logic circuit for function F of example 2.

By using the graphical approach (bubbles etc) we can easily transform the above circuit of fig. 4 into an equivalent one that relies only on NAND gates. Do it if you want. You are experts by now.

<u>Example 3:</u> Find a simplified sum-of-products expession for the prime number detector of example 3 on page 5 of handout # 13.

<u>Answer</u>: <u>Reminder</u>: The prime number detector was stated as: " given a 4-bit input number $(a_3 a_2 a_1 a_0)_2$ the output F is 1 if and only if $(a_3 a_2 a_1 a_0)_2$ is a prime number; otherwise $F = 0$.

We have: $F = \Sigma_{a_3, a_2, a_1, a_0}(1, 2, 3, 5, 7, 11, 13)$.

The corresponding Karnaugh map together with combining the appropriate cells is shown in figure 5 below:

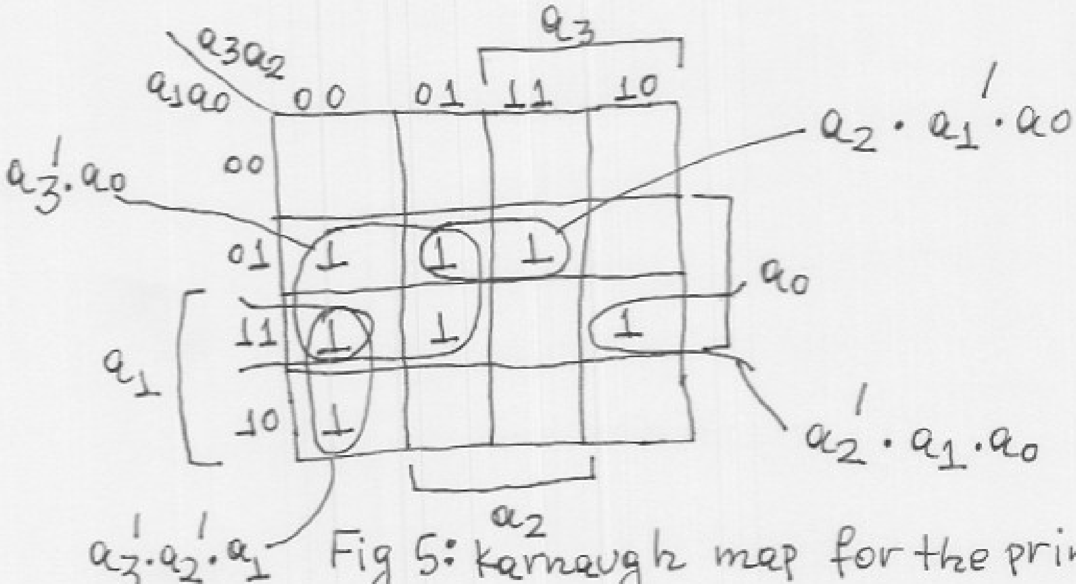


Fig 5: Karnaugh map for the prime number detector.

From the above Karnaugh map we get the simplified sum-of-products expression which is

$$F = a_3' \cdot a_0 + a_3' \cdot a_2' \cdot a_1 + a_2' \cdot a_1 \cdot a_0 + a_2 \cdot a_1' \cdot a_0 \quad (5)$$

Equation (5) above suggests the following minimized AND-OR logic circuit for the prime number detector shown in figure 6 below:

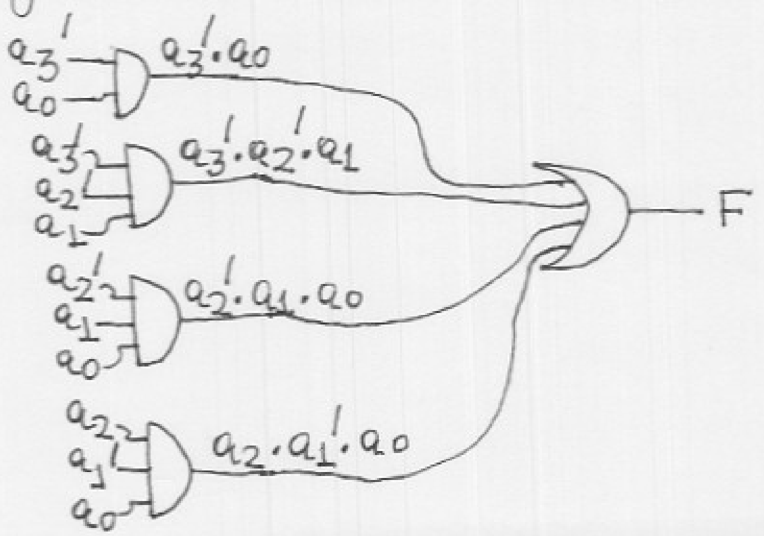


Figure 6: Simplified AND-OR logic circuit for the prime number detector.

Comparing figure 6 of this handout (bottom of previous page) with figure 4 of handout #13, (on page 9 of handout #13; this was the simplified AND-OR logic circuit for the prime number detector that we got <u>algebraically</u>), we see that both figures rely on the same number of gates; (4 AND gates in the first level and one OR gate in the second level). The differences are that the circuit of fig. 4 of handout # 13 relies on one 2-input AND gate and three 4-input AND gates in the first level and one 4-input OR gate in the second level, while the circuit of fig. 6 of this handout relies on one 2-input AND gate and three 3-input AND gates in the first level and one 4-input OR gate in the second level. So the accomplishment is that the circuit of fig. 6 of this handout relies on smaller gates (as compared to this of fig. 4 of handout # 13) which are <u>cheaper</u> and <u>faster</u>. That is something instead of nothing!!

<u>Question</u>: Why did we not get the very simplified expression for F of eq. (5) on previous page (F is the output of the prime # detector) in handout # 13?

<u>Answer</u>: Because we did not apply theorem $X+X=X$ there. We only applied theorem of eq. (1) of page 1 of this handout. When we simplified F using the Karnaugh map method, we used theorem of eq. (1) of page 1 (of course) but we also used the theorem $X+X=X$. Why? Because we combined cell 3 once with cells 1, 5, 7, once with cell 2 and once with cell 11; we also combined cell 5 once with cells 1, 3, 7 and once with cell 13. So we used theorem $X+X=X$ several times in the Karnaugh map method. This is

easy doing it with the Karnaugh map method but not so obvious when you do it algebraically; (it is not so obvious which product terms you should duplicate I mean). So you see how convenient tool the Karnaugh map is? Enjoy it!! Note: In fig. 5 of previous page (the Karnaugh map) we took the product term $a_3' \cdot a_2' \cdot a_1 \cdot a_0$ three times and the product term $a_3' \cdot a_2 \cdot a_1' \cdot a_0$ twice.

An interesting example follows. Why interesting? You will find out below:

Example 4: Find simplified sum-of-products expressions for the logic function $F = \sum_{A,B,C,D}(1,4,5,7,12,14,15)$. Why did I say expressions not expression? Because as you will see below I will find four equally simplified expressions for F.

Answer: I will provide 4 Karnaugh maps. I could have used the same Karnaugh map but that might have been a little bit confusing (not too much). The four Karnaugh maps are shown in figures 7, 8, 9, 10.

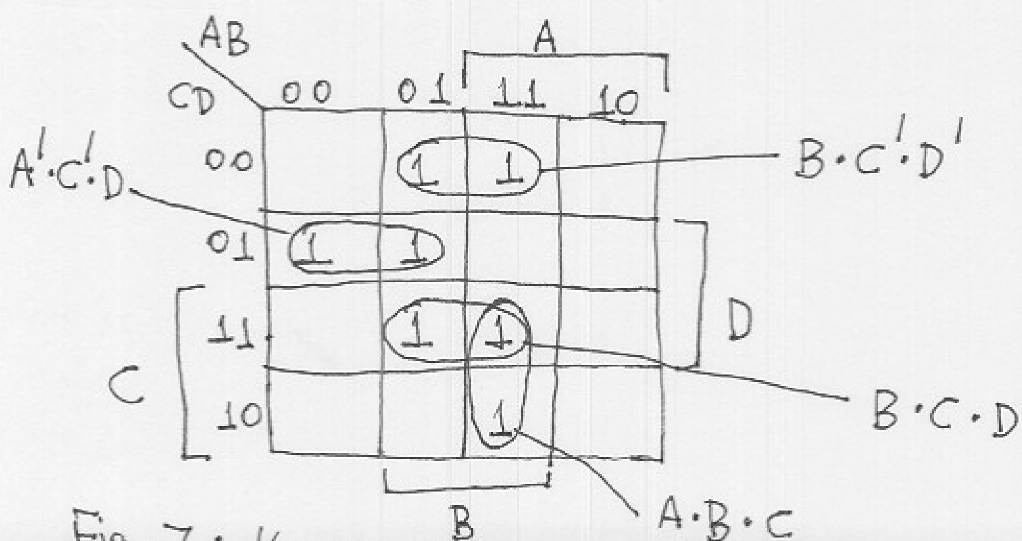


Fig. 7: Karnaugh map for F of example 4.

The Karnaugh map of fig. 7 of previous page gives the following simplified sum-of-products expression for F:

$$F = A' \cdot C' \cdot D + B \cdot C' \cdot D' + A \cdot B \cdot C + B \cdot C \cdot D \quad (6).$$

From eq. (6) above, one can easily get a minimized AND-OR logic circuit for F. Do it if you want. We have done it many times, so by now it should be trivial.

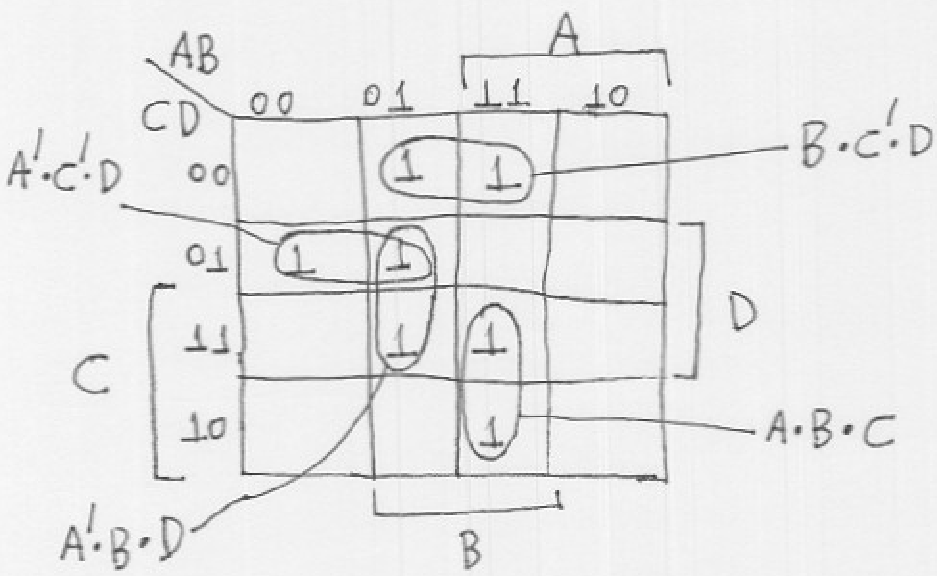


Fig. 8: Karnaugh map for F of example 4.

The Karnaugh map of fig. 8 above gives the following simplified sum-of-products expression for F:

$$F = A' \cdot C' \cdot D + B \cdot C' \cdot D' + A \cdot B \cdot C + A' \cdot B \cdot D \quad (7).$$

From eq. (7) above, one can easily get a minimized AND-OR logic circuit for F. Do it if you want.

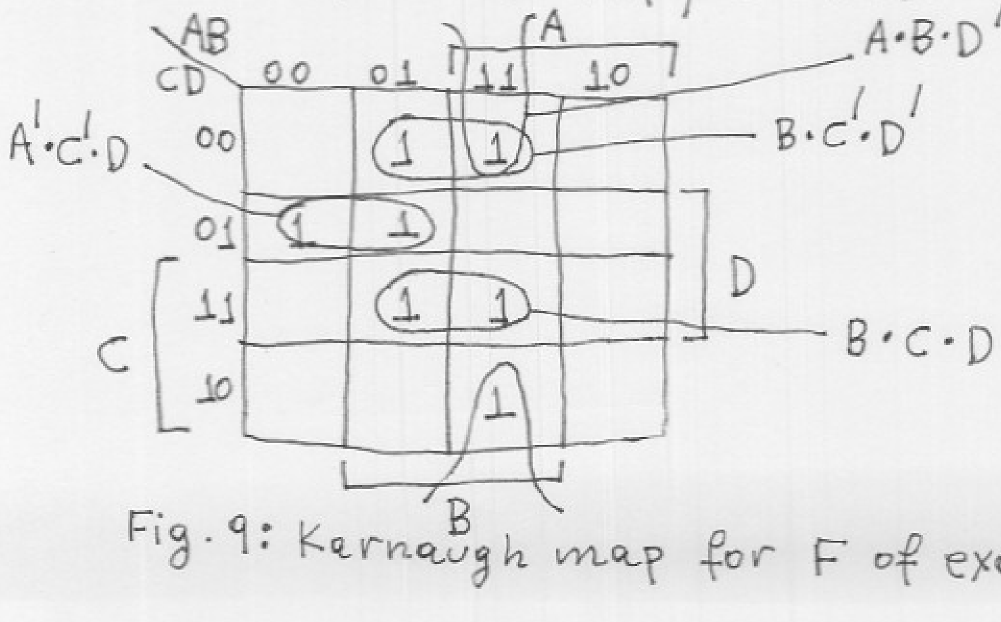


Fig. 9: Karnaugh map for F of example 4.

The Karnaugh map of fig. 9 of previous page gives the following simplified sum-of-products expression for F:

$$F = A' \cdot C' \cdot D + B \cdot C' \cdot D' + A \cdot B \cdot D' + B \cdot C \cdot D \qquad (8)$$

From eq. (8) above, we can easily get a minimized AND-OR logic circuit for F. Do it if you want.

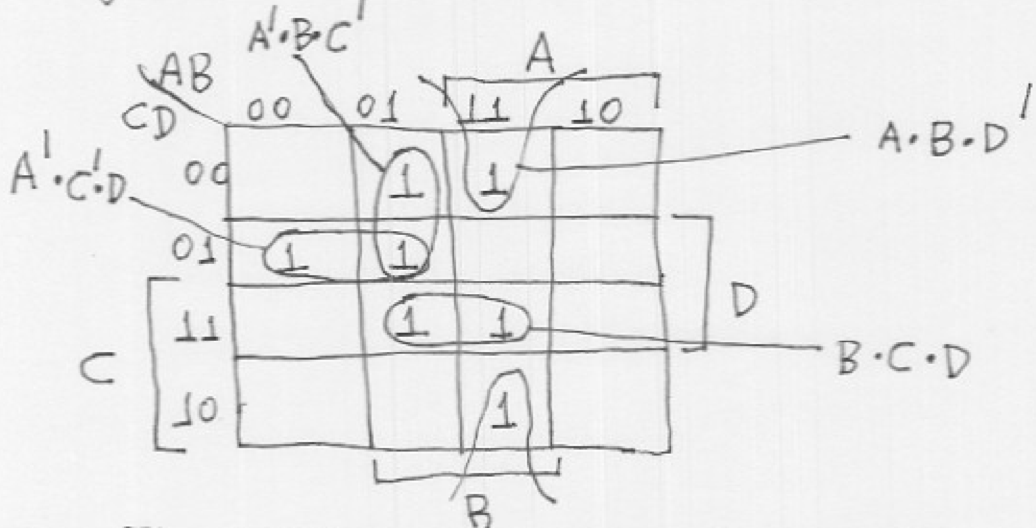


Fig 10: Karnaugh map for F of example 4.

The Karnaugh map of fig. 10 above gives the following simplified sum-of-products expression for F:

$$F = A' \cdot C' \cdot D + A' \cdot B \cdot C' + A \cdot B \cdot D' + B \cdot C \cdot D \qquad (9)$$

From eq. (9) above, one can very easily get a minimized AND-OR logic circuit for F. Do it if you want.

Question: Why did I say earlier that this example is interesting?

Answer: Because we got four different simplified sum-of-products expressions for the same logic function F. So the solution here is not unique. The four simplified expressions that we got are shown in equations (6), (7), (8), (9) and are equally simplified; (they all consist of four product terms, each with three literals). If we realize equations (6), (7), (8), (9) as AND-OR logic circuits, all four logic circuits will rely on four 3-input AND gates in the first level and one 4-input OR gate in the second level.

15

A definition follows:

- Definition: A minimal sum of a logic function $F(X_1, X_2, \ldots, X_n)$ is a sum-of-products expression for F such that no sum-of-products expression for F has sion with the same number of product terms has at least as many literals. In other words, (to make it more clear), the minimal sum has the fewest possible product terms (first-level gates and second-level gate inputs) and, within that constraint, the fewest possible literals (first-level gate inputs).

Note: The subject of Karnaugh maps will be continued in the next handout. The subject is very important. And very convenient and interesting I think. What do you think?