KARNAUGH MAPPING (k-mapping)

- the derivation of the Boolean expression is focused on cells.

The type of k-map to use is dependent on the number of inputs and how many k-maps to make is dependent on the number of outputs given a problem statement. Say if the problem requires three inputs and four outputs, then we will use the four (outputs) maps for the three variable map (inputs)

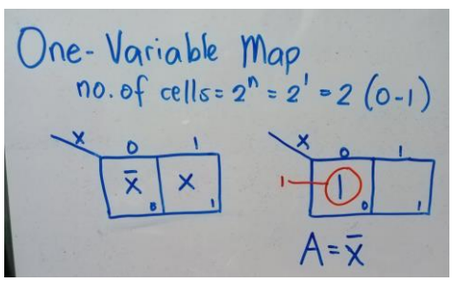
Generally, in every type of map, the number of cells is computed as 2^n, where n is the number of inputs. The type of map generally signifies the number of inputs; example the one-variable k-map is use for a solution requiring one input. The two-variable k-map is use for a solution requiring two inputs, and so on. We will use up to the five variable map only.

For our discussions on Boolean algebra, we are going to use the Sum of Products (SOP) expressions. With that we are going to collect minterms (1) from the truth table to the maps. Unlike the (Product of Sums) POS which requires the use of maxterms (0). Note that SOP and POS were discussed in the ppt file on Boolean algebra.

To simplify, we have to group the minterms by powers of 2, meaning by 1, 2, 4, 8, 16, and 32 members only per group depending on the placement of the minterms. You cannot by 3, 5, 6 ,7 or any other number when they are not on the powers of 2. We can group minterms if they are adjacent (neighbors) horizontally or vertically, but not diagonally and each group should be the maximum number of minterms possible. It is also important that no minterms should be left ungrouped.

Types of K-maps

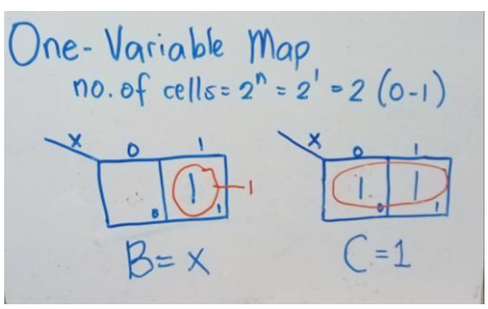
1. ***One Variable Map***



In this map, we will use x to represent the input. So x can be normal value x or the negated x, represented by x' or x with a bar above it (called as x-bar). So if you will notice that there are column headings 0 and 1, mainly because a variable can hold either a logic 0 or a logic 1 value, because we are dealing with Boolean values. So each variable in whatever type of map, holds either a logic 0 or a logic 1.

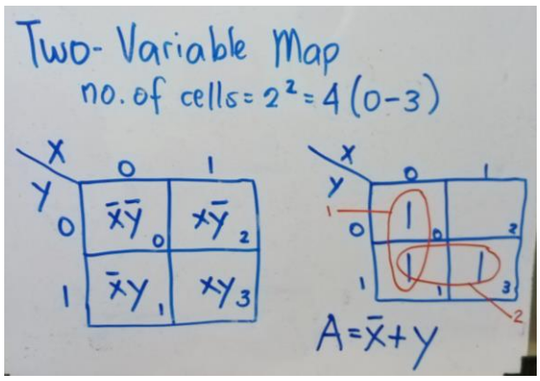
As seen on this example (map at the left side), for the cell locations note that locations are 0 and 1, so a 0 value means the negated value of the variable and the 1 means the normal value of the variable. Why do have cell locations 0 and 1? Because a single valued binary digit can be 0 or 1.

On the map (right side), say we have a minterm at location 0, so we will have a group for this minterm. Since that is representing the negated value, so therefore A (the output for this example) is equal to x' or x-bar.



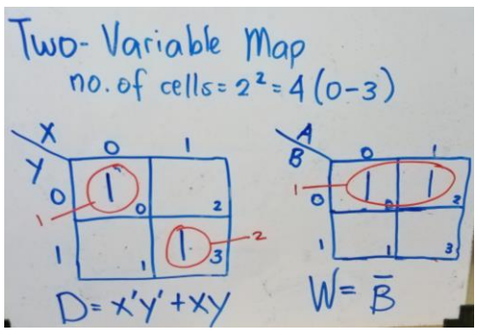
Another example (left side), if B is the output variable and 1 is found at location 1, we have a group with a single member with B = x, since x is in the 1 heading which represents the normal value for x. For the other example (right side), we have a single group with one group but with two minterms, giving the output C, a value of 1. Take note that if a k-map is full of minterms, the output will be equal to 1, such as in this case.

2. *Two Variable Map*



There are four cells for a two variable map. We have two inputs x and y in this case. On the left side are the product term for each location. The column heading represents the value for x (*Most Significant Bit* (MSB) in this case) and the row heading represents the value for y (*Least Significant BIt* (LSB) in this case). For the cell location number, you need memorize but you have to be familiar with the binary to decimal conversion and vice versa. Note at location 2, the value is 2 because x=1, y=0; in binary xy is 10, and 10 in binary converted to decimal is 2. Why is named location 1, because xy=01 in binary, when converted is 1. Note we need to know the locations because we need this for the simplification.

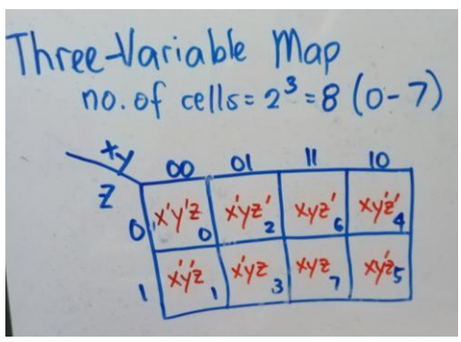
Look at the example (right side), the maximum number of minterms per group should be by 2, and not by 1 because we need to have the highest possible number of minterms per group. Note also that they are grouped vertically and/or horizontally, but not diagonally. How did we derive this expression? First consider group 1, x is on heading 0, that means x-bar (x') but y cancels out because y is both represented in 0 and 1. Remember if a certain variable is represented with both 0 and 1, it cancels out (remember negation law). Now consider group 2, x cancels out because it is both represented in 0 and 1 columns but it is only represented as 1 in y. Therefore, the final Boolean expression x' + y.



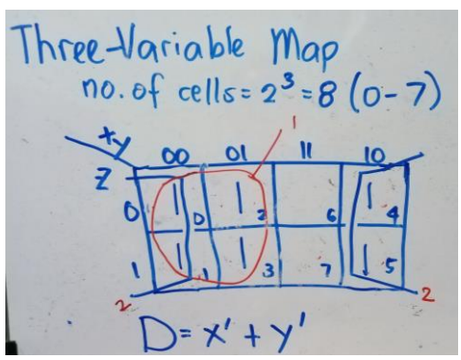
In this example (left side), since we cannot group diagonally, so the minterms must be grouped individually. For the first group x=0 and y=0, therefore we have x'y'. For the second group, x=1, y=1, therefore that is xy. Thus the final expression for the output D Is D=x'y' + xy

In the (right side) example, why do we have W=B(bar) or W=B'? It is because we will have one group with two minterms. A cancels out but B is B=0. Thus W=B'.
Take note here A is the MSB and B is the LSB for the inputs.

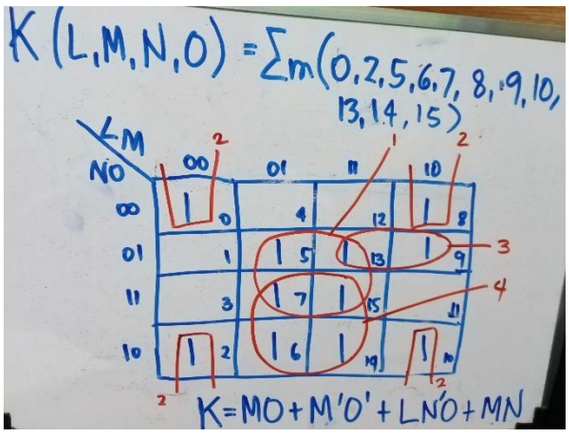
3. *Three Variable Map*



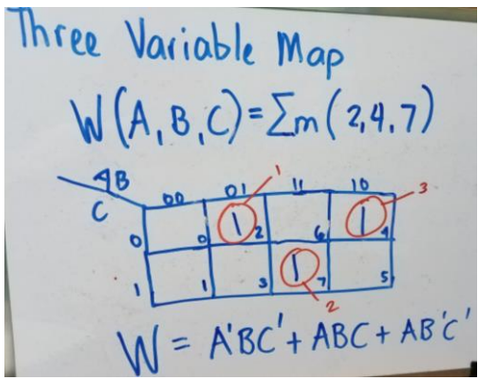
In a three variable map, we have 8 cells (locations 0 to 7). In this example, we have inputs x, y, and z where x is the MSB and Z as the LSB. So take note you read the variable sequence as xyz. In this map, the first bit in the column heading represents x, the second bit in the column heading represents y and the row heading represents z. The equivalent product terms for each location are written in this map. Again for the cell locations, you need not memorize but you have to convert to get the proper location. Example in location 3, xyx = 011 in binary is 3 in decimal. With regards to product term that is x'yz.



In this example, we use the concept of folding. You can fold maps horizontally and/or vertically to form adjacent neighbors of minterms. Group 1 is a group of four minterms. In this comparing x from the two columns will have a common 0 value, which means an x'; reading through y, y cancels out, reading through z, z cancels out. Again remember that for any variable that has representations of both 0 and 1, it cancels out. For group 2, group of four minterms using folding, reading through x, x cancels out; reading through y, y is represented with 0, meaning y' and reading through z, z cancels out. So we have D= x' + y'.

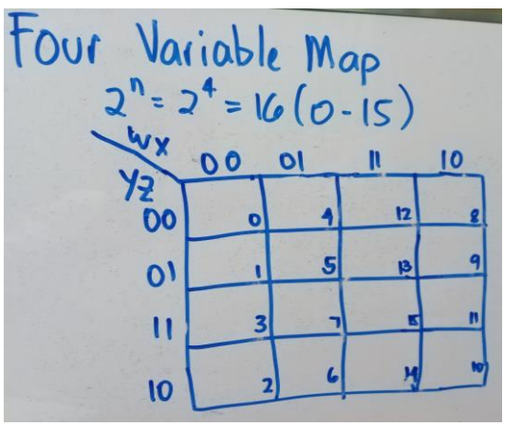


Starting with this example, we are going to solve k-maps in this manner. Take a look at F(x,y,z) = Σm (1,3, 6, 7). What is the significant of this equation? But we have to get to know the parts of this equation. F is the output variable; x, y, z are the inputs which means there are three inputs; Σm means summation of minterms, meaning we are going to collect minterms, hence we are to form SOP. If you happen to come across ΣM, meaning you have to collect maxterms and form the POS expression. So in this case, we are going to form SOP; 1,3,6,7 are the cell locations where the minterms are located. So if you notice, we plot minterms (1) on locations 1,3,6,7. In this example, although we have four minterms, we cannot group by 4, hence we group them by 2 in two groups. Take note that there is no need to regroup minterms at locations 3 and 7 because all minterms have their groups already. So part of the analysis also is that you need to see to it that grouping minterms should not be repeated when it unnecessary. For group 1, x is common at 0 meaning x', y cancels out, z is common at 1 meaning z, therefore forming x'z for group 1. For group 2, x=1 meaning z; y is 1 meaning y, and z cancels out, giving xy for group 2. Thus, F = x'z + xy.

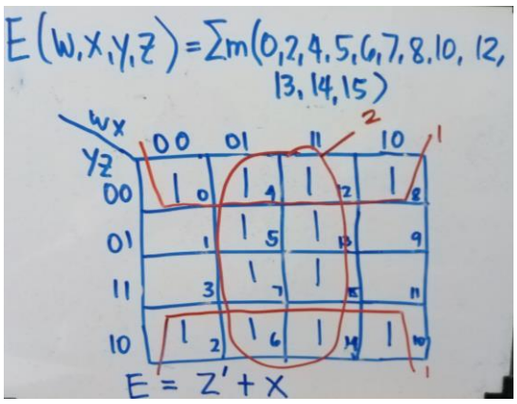


For this example, groupings must be done individually, since we cannot group diagonally. Study why we have the output A to be as written on the example.

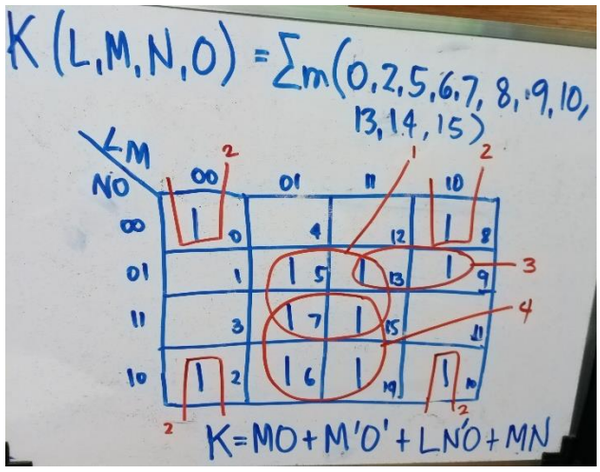
4. Four Variable Map



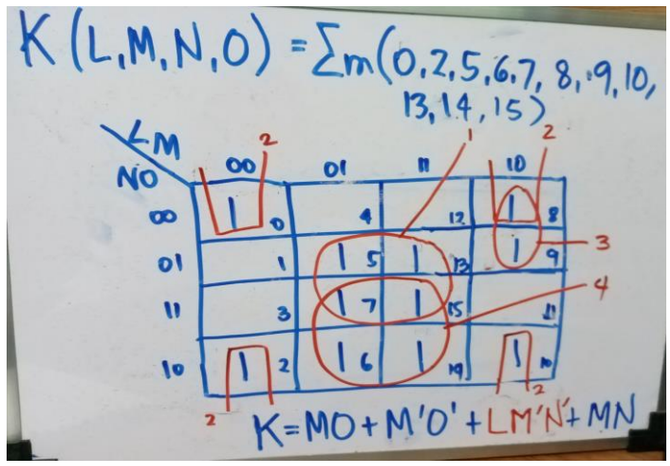
The four variable map means there are 4 inputs. In this case we use wxyz with w as the MSB and z as the LSB. The first digit in the column heading represents w, the second digit represents x. In the row heading the first digit represents y and the second digit represents z. Hence for location 10, that is wxyz=1010, which means an equivalent product term for wx'yz'. As part of your quiz for this lecture, you have to complete the list of product terms inside the map (you may refer to the format in the three variable map)

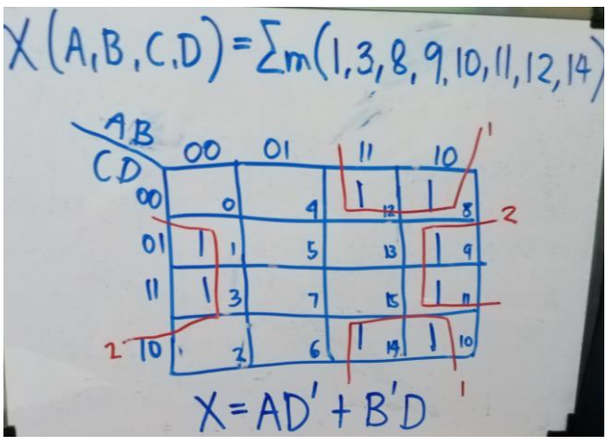


In this example, the output is E with minterms plotted on the locations as state din the problem. You can use the concept of folding for the four (and five) variable map as well. In this case we group the minterms by 8 for each group. So for group 1, reading from w, w cancels out; x cancels out, y cancels out, z has zero representation meaning z'. For group 2, w cancels out, x has representation in 1, meaning x; y cancels out, z cancels out. So putting the equation together, we have E= z' + x.



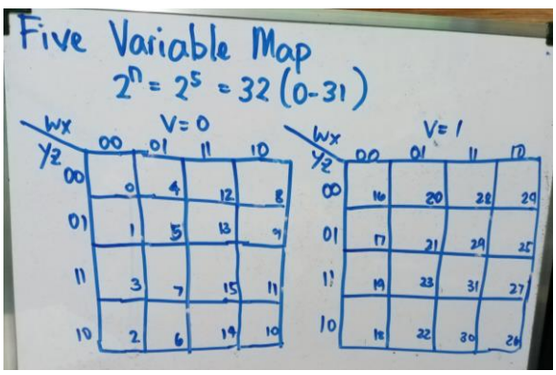
In this example, you may have different groupings other than represented. You can form four groups for this map, I hope that you can get the final expression as presented upon analyzing the map. Now I want you to focus on group 3, the minterms at location 9 and 13 are paired. But you can also group minterms at location 8 and 9 to form group 3, such as shown on the next figure below (product term written in red). You also form group 4 in other way, the solution of this will form part of your quiz for this chapter.

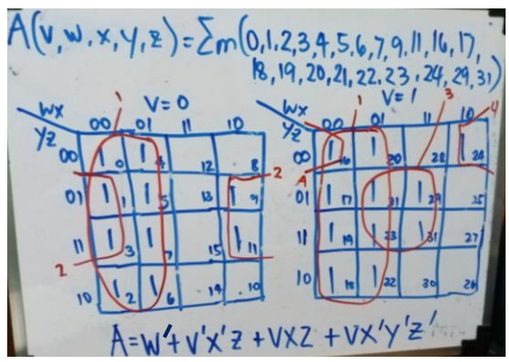




Study this example why the groupings are done in this manner. Trace through to understand the final expression.

5. *Five Variable Map*





The five variable consists of 32 cells, with locations from 0 to 31. If you notice, a five variable map consists of two four-variable map. With the first map for the locations 0 to 15 and the second map for the locations 16 to 31. The difference is that the first map represents when the MSB is equal to 0 (in this case, v=0) and the second map is when the MSB is equal to 1 (V=1). In this map we read the variable sequence as vwxyz, with v as the MSB and z as the LSB. Example at location 15, that is vwxyz = 01111, or 15 in decimal. Considering location 23, that is vwxyz=10111 or 23 in decimal.

In this type of map, you can use folding and mirroring. Think of the mirroring as if persons are waiting in a queue line where the person second person sees the back of the first person. In this case, the second map can be aligned with the first map given that they are one after the other and not face to face. As an example, we have the example at the right figure above. Group 1 consists of 16 minterms with 8 from map1 and 8 from map2 which are actually in mirror with each

other; Group 2 made use of the folding in map1; Group 3 consists of 4 minterms in map2; and group 4 uses folding in map2. For the simplification read first from v all the way to z.

In group 1: v cancels out because there are representations in both v=0 and v=1; w has 0 representation, meaning w' while x, y, and z cancel out, leaving w' for group 1.
For group 2: v is 0 meaning v', w cancels out, x is 0 meaning x', y cancels out, and z is 1, meaning z. Thus forming the product term v'x'z for group 2.
For group 3: the product term is vxz because v=1 meaning v, w cancels out, x is 1 meaning x, y cancels out, and z is 1, leaving z.
For group 4: the product term is vx'y'z' because v=1 meaning v, w cancels out, x=0 meaning x', y=0 meaning y', and z=0 meaning z'.

Take note that we may have different group numberings, the commutative law applies. Just take note of the do's and don'ts in groupings minterms. Next part will be on logic gates. Then after we will apply the simplification and logic gates discussion in circuit designing.