# K-Maps: A Visual Tool for Systematic Design of Combinational Circuits Using Multiplexers.

**M.V.Shuma-Iwisi**
Department of Electrical Engineering
University of Botswana
Gaborone, Botswana

*One of the application areas of digital multiplexers is in implementation of logic equations. Truth tables can be used as a design tool, but they become very limited when design details are explored. It will be shown in this paper that the Karnaugh map ( K-map) is an essential tool in the systematic detailed design of combinational circuits using multiplexers. The K-Map will be used in two ways. First as a visual tool and secondly as a tool for decision making criteria.*

## 1. INTRODUCTION

The Karnaugh map is a graphical device used for simplifying logic equations or to convert a truth table to its corresponding logic circuit in a simple orderly process. It can also be defined as a means for showing the relationship between logic inputs and desired outputs [1]. The K-map can be used for problems involving any number of input variables, but its practical usefulness is limited to problems with a maximum of five input variables.

The multiplexer is a medium scale integrated logic circuit defined generally as having $n$ select lines, $2^n$ data input lines and a single output line. An enable input is always available on a multiplexer for purposes of chip control. Most available multiplexer Integrated Circuit's (IC's), have the number of select lines $n$ ranging from 1 to 4.

When a multiplexer is used for generation of logic equations, two design cases evolve depending on the multiplexer available for the implementation.

*Case 1:*
The number of select lines $n$ on the multiplexer to be used is equal to the number of input variables $m$, of the logic equation to be generated.

*Case 2:*
The number of select lines $n$ on the multiplexer to be used is less than the number of input variables $m$ of the logic equation to be generated.

In case one, as will become apparent, the K-Map becomes a visual tool as each multiplexer data input is related to the respective min-term on the K-Map. In case 2, the K-Map is used both as a visual and a decision making tool through simplification. This function is seen as select lines are associated with the inputs of the logic equation, and corresponding multiplexer data input areas are developed on the K-Map. This paper will focus only on case 1.

## 2. IMPLEMENTATION OF LOGIC EQUATIONS WITH M INPUT VARIABLES USING MULTIPLEXERS WITH N SELECT LINES WHERE $m = n$.

If $m = n$, then the number of input combinations for which the logic equation is defined is equal to the number of data input lines available on the multiplexer. The individual K-Map cell entries for the logic equation, represent the logical levels on each of the corresponding data input lines on the multiplexer.

It is important to note that the order of assignment of the logic input variables to the select lines of the multiplexer gives several K-Map cell assignments to multiplexer data input lines. This is where the K-Map becomes a necessary visual tool, to visualize which K-Map cell corresponds to which data input line on the multiplexer. Most design errors made are typically a problem of order of assignment of input variables to select lines and the visual interpretation of the cell assignments.

Some one might question the importance of the order of assignment of the input variables to select lines. First remember that the function table of a particular multiplexer, binds the designer to use it in a specific way. For example if we take the function table for the multiplexer SN74LS151, an 8 to 1 line multiplexer [2], we will find out that the order of significance of the select lines is such that select line $C$ is the most significant select line, and select line $A$ is the least significant select line. This implies that only combinations of $CBA$ in the specified order will select the correct data input line on the multiplexer.

Secondly in any logic equation, when input variables are assigned, an order of significance is attached to each input variable. For example a logic equation defined as $F(X, Y, Z) = \Sigma m( 1, 2, 5, 6, 7 )$ implies that $X$ is the most significant input variable and Z is the least significant input variables. Once the assignment is done by the designer, then the order of significance of the defined input variables, should remain fixed to ensure consistency in the design process.

In the discussions to follow, we will use the following abbreviations for multiplexers:
For a multiplexer with $n$ select lines, there will be $2^n$ data input lines abbreviated as $I_{2^n-1}, I_{2^n-2}, \ldots\ldots I_0$, and $n$ select lines $S_{n-1}, \ldots\ldots S_0$, where $S_{n-1}$ will be the most significant select line and $S_0$ will be the least significant select line. For example a multiplexer with 3 select lines will have 8 data input lines abbreviated as $I_7$, $I_6$, $I_5$, ------, $I_0$ and select lines $S_2$, $S_1$, and $S_0$.

In the case of logic equations we will define logic equations with $m$ input variables and $2^m$ input combinations. The letters *A, B, C, D* etc. will be used to define the input variables. *A* will always correspond to the most significant input variable and *D* for a four variable or *C* for a three variable logic equation, will correspond to the least significant input variable. For example a 3 variable logic equations will have 8 input combinations and the input variables will be *A, B, C,* where *A* is the most significant variable.

We will now proceed to demonstrate the correct design process for case 1, by way of examples of different order of assignments. First a multiplexer with 3 select lines and a logic equation with 3 input variables will be used. Secondly, a multiplexer with four select lines and a four variable logic equation will be used. It is important to note that for an $m$ variable logic equation there a $m! = m \times (m-1) \times (m-2) \times \ldots\ldots \times 1$ possible assignments.

## 3. IMPLEMENTATION OF A 3 VARIABLE LOGIC EQUATION USING A MULTIPLEXER WITH 3 SELECT LINES.

In the case where $m = 3$, there are 6 possible different assignments as shown in table 1. Three assignments marked with * will be used in the examples to follow.

| | | |
|---|---|---|
| $A \rightarrow S_2$ | $B \rightarrow S_1$ | $C \rightarrow S_0$* |
| $A \rightarrow S_2$ | $C \rightarrow S_1$ | $B \rightarrow S_0$ |
| $B \rightarrow S_2$ | $A \rightarrow S_1$ | $C \rightarrow S_0$* |
| $B \rightarrow S_2$ | $C \rightarrow S_1$ | $A \rightarrow S_0$ |
| $C \rightarrow S_2$ | $A \rightarrow S_1$ | $B \rightarrow S_0$ |
| $C \rightarrow S_2$ | $B \rightarrow S_1$ | $A \rightarrow S_0$* |

Table 1: Possible Assignments for a 3 variable logic equation.

### 3.1 Example 1: Order of Assignment: $A \rightarrow S_2$, $B \rightarrow S_1$, and $C \rightarrow S_0$.

This is the assignment we will call the ***"fundamental assignment"***. In this assignment, the most significant variable is assigned to the most significant select line. All the other variables are assigned to the respective select line in an ascending order, finally the least significant variable is assigned to the least significant select line.

In each assignment two K-Maps emerge. One K-Map to portray the logic equation minterms positions in the K-Map and the respective contents. This we will call the ***"Function map"***. The second K-Map portrays the assignment result, we will call this the ***"Assignment map"***. The assignment map corresponding to the fundamental assignment will be called the ***"fundamental assignment map"***. The fundamental assignment map will form the basis of comparison for all the other possible assignments for a particular implementation. The assignment map will automatically define the logic levels for each data input of the multiplexer. The logic levels are the actual cell entries.
The logic equation to be used as an example is
$F(A, B, C) = \Sigma m ( 1, 2, 5, 6, 7 )$. Figure 1 below is the function map for the given example, while figure 2 is the fundamental assignment map.

| A \ BC | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |

Fig. 1: Function Map

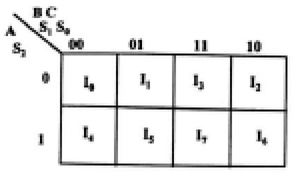


Fig. 2 : Assignment map 1.

Using the two K-Maps one can make the following conclusion for this particular assignment that $I_{2^n-1}$ = F(A, B, C) at minterm (m) $_{2^n-1}$. Hence $I_0 = 0$, $I_1 = 1$, $I_2 = 1$, $I_3 = 0$, $I_4 = 0$, $I_5 = 1$, $I_6 = 1$ and $I_7 = 1$.

### 3.2 Example 2: Order of Assignment: $C \rightarrow S_2$, $B \rightarrow S_1$, $A \rightarrow S_0$

The function map for any assignment for a defined logic equation is the same. However the assignment map is different for different assignments. We shall explain this first by use of truth tables. Table 1 depicts the assignment in example 1, and table 2 represents the assignment in example 2. :

| $S_2$ | $S_1$ | $S_0$ | Minterm | Data Input selected |
|---|---|---|---|---|
| ↓ | ↓ | ↓ | | |
| A | B | C | $m_n$ | |
| 0 | 0 | 0 | $m_0$ | $I_0$ |
| 0 | 0 | 1 | $m_1$ | $I_1$* |
| 0 | 1 | 0 | $m_2$ | $I_2$ |
| 0 | 1 | 1 | $m_3$ | $I_3$* |
| 1 | 0 | 0 | $m_4$ | $I_4$ |
| 1 | 0 | 1 | $m_5$ | $I_5$ |
| 1 | 1 | 0 | $m_6$ | $I_6$* |
| 1 | 1 | 1 | $m_7$ | $I_7$ |

Table2: Assignment 1

| $S_2$ | $S_1$ | $S_3$ | Minterm | Data Input Selected |
|---|---|---|---|---|
| ↓ | ↓ | ↓ | | |
| C | B | A | $m_n$ | $I_N$ |
| 0 | 0 | 0 | $m_0$ | $I_0$ |
| 1 | 0 | 0 | $m_4$ | $I_4$* |
| 0 | 1 | 0 | $m_2$ | $I_2$ |
| 1 | 1 | 0 | $m_6$ | $I_6$* |
| 0 | 0 | 1 | $m_1$ | $I_1$ |
| 1 | 0 | 1 | $m_5$ | $I_5$ |
| 0 | 1 | 1 | $m_3$ | $I_3$* |
| 1 | 1 | 1 | $m_7$ | $I_7$ |

Table 3: Assignment 2

On inspecting the truth tables, one can see that the new assignment has brought about changes in the marked minterms in the assignment map as shown below in figure 3 i.e. $m_1$ in the fundamental assignment map corresponds to $m_4$ in example 2, $m_3$ to $m_6$ and $m_6$ to $m_3$.

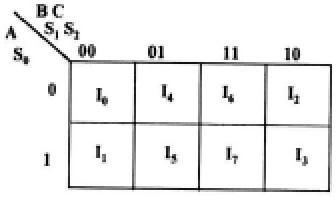


Fig. 3: Assignment Map 2.

Matching assignment map in figure 3 to the function map in figure1 the following conclusion is reached. $I_0 = 0$, $I_1 = 0$, $I_2 = 1$, $I_3 = 1$, $I_4 = 1$, $I_5 = 1$, $I_6 = 0$, $I_7 = 1$. The two solutions for the two assignments are not equivalent. It is important to note that if the assignment map for example 1 is used for the assignment in example 2, the resulting conclusion is $I_0 = 0$, $I_1 = 1$, $I_2 = 1$, $I_3 = 0$, $I_4 = 0$, $I_5 = 1$, $I_6 = 1$, $I_7 = 1$. This is an incorrect conclusion and the deduced circuit will not work!

### 3.3 Example 3: Order of assignment: $B \rightarrow S_2$, $A \rightarrow S_1$, $C \rightarrow S_0$.

In the same manner as shown in example 2, the resulting assignment map is given below in figure 4. Note the assignment change in this case affects $m_4$, $m_2$, $m_3$, and $m_5$ when compared to the fundamental assignment map.

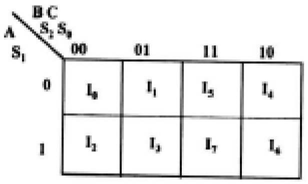


Fig. 4: Assignment Map 3.

Matc
hing the assignment map in figure 4 to the function map in figure 1, the result is: $I_0 = 0$, $I_1 = 1$, $I_2 = 0$, $I_3 = 1$, $I_4 = 1$, $I_5 = 0$, $I_6 = 1$, $I_7 = 1$.

## 4. IMPLEMENTATION OF A FOUR VARIABLE LOGIC EQUATION USING A MULTIPLEXER WITH 4 SELECT LINES.

We shall use the logic equation defined as $F(A, B, C, D) = \Sigma m(0, 3, 5, 6, 9, 10, 11, 13)$ as the demonstration example. In this case where $n=m=4$ there are 24 possible assignments. The four assignments to be used are given in table 4 below.

| | | | |
|---|---|---|---|
| $A \rightarrow S_3$ | $B \rightarrow S_2$ | $C \rightarrow S_1$ | $D \rightarrow S_0$ |
| $B \rightarrow S_3$ | $C \rightarrow S_2$ | $D \rightarrow S_1$ | $A \rightarrow S_0$ |
| $C \rightarrow S_3$ | $D \rightarrow S_2$ | $A \rightarrow S_1$ | $B \rightarrow S_0$ |
| $D \rightarrow S_3$ | $A \rightarrow S_2$ | $B \rightarrow S_1$ | $C \rightarrow S_0$ |

Table 4: A selection of possible assignments for $n=m=4$.

### 4.1 Example 4: Order of Assignment: $A \rightarrow S_3$, $B \rightarrow S_2$, $C \rightarrow S_1$, $D \rightarrow S_0$.

This is the fundamental assignment. As in the other examples, we will develop a function map and an assignment map. The two will be used to determine the solution. For the logic equation in use, the two maps are given below.

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 1 | 0 |
| 01 | 0 | 1 | 0 | 1 |
| 11 | 0 | 1 | 0 | 0 |
| 10 | 0 | 0 | 1 | 1 |

Figure 5: Function Map 2

If we compare corresponding map entries on the two maps, the conclusion is: $I_0 = 1$, $I_1 = 0$, $I_2 = 0$, $I_3 = 1$, $I_4 = 0$, $I_5 = 1$, $I_6 = 1$, $I_7 = 0$, $I_8 = 0$, $I_9 = 0$, $I_{10} = 1$, $I_{11} = 1$, $I_{12} = 0$, $I_{13} = 1$, $I_{14} = 0$, $I_{15} = 0$.

| A B / $S_3$ $S_2$ \ C D / $S_1$ $S_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | $I_0$ | $I_1$ | $I_3$ | $I_2$ |
| 01 | $I_4$ | $I_5$ | $I_7$ | $I_6$ |
| 11 | $I_{12}$ | $I_{13}$ | $I_{15}$ | $I_{14}$ |
| 10 | $I_8$ | $I_9$ | $I_{11}$ | $I_{10}$ |

Figure 6:Assignment map 4.

### 4.2: Example 5: Order of Assignment: $B \rightarrow S_3$, $C \rightarrow S_2$, $D \rightarrow S_1$, $A \rightarrow S_0$.

The assignment map for this case is given below. It can be noted that with the exception of minterms $m_0$ and $m_{15}$ all other entries have changed locations, when this assignment map is compared to the fundamental map in figure 6. In the resulting assignment map, notice that all K-Map basic rules have been preserved.

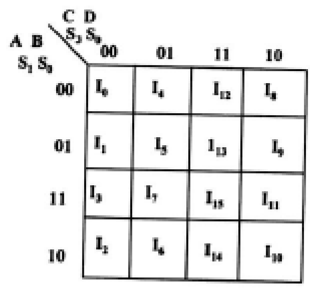
| A B / $S_0$ $S_3$ \ C D / $S_2$ $S_1$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | $I_0$ | $I_2$ | $I_6$ | $I_4$ |
| 01 | $I_8$ | $I_{10}$ | $I_{14}$ | $I_{12}$ |
| 11 | $I_9$ | $I_{11}$ | $I_{15}$ | $I_{13}$ |
| 10 | $I_1$ | $I_3$ | $I_7$ | $I_5$ |

Figure 7:Assignment map 5.

The result obtained when corresponding entries on function map 2 and assignment map 5 are equated is the following: $I_0 = 1$, $I_1 = 0$, $I_2 = 0$, $I_3 = 0$, $I_4 = 0$, $I_5 = 1$, $I_6 = 1$, $I_7 = 1$, $I_8 = 0$, $I_9 = 0$, $I_{10} = 1$, $I_{11} = 1$, $I_{12} = 1$, $I_{13} = 0$, $I_{14} = 0$, $I_{15} = 0$.

### 4.3: Example 6: Order of Assignment: $C \rightarrow S_3$, $D \rightarrow S_2$, $A \rightarrow S_1$, $B \rightarrow S_0$.

The corresponding assignment map for this assignment is given in figure 8, and as in the previous examples, we will use the function map and the assignment map to visualize the correct logic levels for each input of the multiplexer.



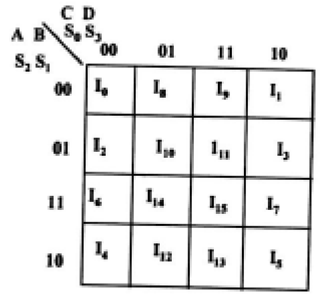
| AB $S_1 S_0$ \ CD $S_3 S_2$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | $I_0$ | $I_4$ | $I_{12}$ | $I_8$ |
| 01 | $I_1$ | $I_5$ | $I_{13}$ | $I_9$ |
| 11 | $I_3$ | $I_7$ | $I_{15}$ | $I_{11}$ |
| 10 | $I_2$ | $I_6$ | $I_{14}$ | $I_{10}$ |

**Figure 8:Assignment map 6.**

In this particular assignment we notice that unlike in example 4.2, where all minterms except $m_0$ and $m_{15}$ changed positions on the assignment map, in this case all but minterms $m_0$, $m_5$, $m_{10}$ and $m_{15}$ have changed positions, when the assignment map 6 is compared to the fundamental assignment map in figure 6. It is important to note that each specific assignments brings about specific changes on the minterm positions. It can be concluded therefore that the correct definitions for each input of the multiplexer will be: $I_0 = 1$, $I_1 = 0$, $I_2 = 0$, $I_3 = 0$, $I_4 = 0$, $I_5 = 1$, $I_6 = 0$, $I_7 = 1$, $I_8 = 0$, $I_9 = 1$, $I_{10} = 1$, $I_{11} = 0$, $I_{12} = 1$, $I_{13} = 0$, $I_{14} = 1$, $I_{15} = 0$.

### 4.4: Example 7: Order of Assignment: $D \rightarrow S_3$, $A \rightarrow S_2$, $B \rightarrow S_1$, $C \rightarrow S_0$.

The corresponding assignment map is given in figure 9. If this assignment map is compared to the fundamental assignment map, it can be deduced that all other minterms change their positions except $m_0$ and $m_{15}$.



| AB $S_2 S_1$ \ CD $S_0 S_3$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | $I_0$ | $I_8$ | $I_9$ | $I_1$ |
| 01 | $I_2$ | $I_{10}$ | $I_{11}$ | $I_3$ |
| 11 | $I_6$ | $I_{14}$ | $I_{15}$ | $I_7$ |
| 10 | $I_4$ | $I_{12}$ | $I_{13}$ | $I_5$ |

**Figure 9:Assignment map 7.**

The inputs of the multiplexer in this assignment can be correctly defined as before to be: $I_0 = 1$, $I_1 = 0$, $I_2 = 0$, $I_3 = 1$, $I_4 = 0$, $I_5 = 1$, $I_6 = 0$, $I_7 = 0$, $I_8 = 0$, $I_9 = 1$, $I_{10} = 1$, $I_{11} = 0$, $I_{12} = 0$, $I_{13} = 1$, $I_{14} = 1$, $I_{15} = 0$.

## 5. CONCLUSIONS

In all the examples discussed, the function map remains the same. However the assignment map changes depending on the assignment made when compared to the fundamental assignment map. The solution is determined by equating cell entries on the function map to corresponding cell entries on the assignment map. The K-Map becomes a visual tool to the designer.

It is important to note that, when a multiplexer is used in implementation of a logic equation where $n = m$, then the designer is advantaged. First because there is no need of simplifying the equation, and secondly because the design acquires the speed performance of the multiplexer IC being used [3].

Multiplexers become simpler and better implementation devices compared to logic gates because of the elimination of propagation delays that are dependent on the number of stages in the implementation. Multiplexers when compared to programmable array logic devices, are simpler implementation devices because folding, factoring and partitioning [4] do not have to be considered in the design process. Further more on speed consideration they are faster than programmable logic arrays [4]. The only disadvantage is the fact that one cannot implement multiple output equations. As has been shown, a simple and systematic design methodology has been successfully demonstrated.

## 6. REFERENCES

1. Tocci, R. J. Digital Systems Principles and Applications, Fifth Edition, Prentice Hall, Englewood Cliffs New Jersey; 1991.

2. Standard TTL, Low-Power Schottky, Schottky, Data Book, Volume 1, Texas Instruments, 1989.

3. Hayes, J. P. Introduction to Digital Logic Design, Addison-Wesley Publishing Company, Inc. 1993.

4. Daniels, J. D. Digital Design from Zero to One, John Wiley & Sons, Inc., New York, 1996