B-4.

Karnaugh Maps

OBJECTIVEs:

In this experiment, the student will design, build, and test an Invalid BCD Code Detector and a three input XOR gate that have been simplified by means of a Karnaugh Map.

READING:

MATERIALS NEEDED:

74 LS 04 1 EA

74 LS 32 1 EA

74 LS 08 2 EA

LED 1 EA

330Ω 1 EA

POWER SUPPLY

SUMMARY OF THEORY:

Previously, it was observed that it is possible to minimize the number of gates that are required to implement a logic function through the use of Boolean Laws and Rules and DeMorgan's Theorem. It is important to reduce a logic function to the minimum number of gates that will generate an equivalent function to reduce a circuits cost and size and increase its reliability. It is tedious and often difficult to use these Boolean reduction techniques since it requires much practice and ingenuity on the part of the circuit designer.

There is a more direct approach that may be used simplify the Boolean reduction process. This method is called the Karnaugh Map, which uses a systematic approach to reduce a Boolean Equation to its simplest Sum-of-Products (SOP) form. This is accomplished by creating an arrangement of adjacent cells each of which represents one combination of variables in SOP form for a Boolean Equation. The number of cells for the K-Map is 2n where n is the number of input variables. If we have four variables in a Boolean Equation, A, B, C, and D, there will be 24 or 16 cells in the K-Map.

An adjacent cell is defined as a cell that differs from its neighbor by only a single variable. When we group adjacent cells, we must combine the cells in groups that will include the most variables in powers of 2 and every 1 in the map must be included at least once. Overlapping groups may be made provided they include a 1 that is not included in another group.

To write the simplified SOP form of the Boolean Equation after the 1's on the K-Map have been circled:

1. Write an equation for each group of variables by eliminating those variables that appear in the uncomplemented and complemented form. Each group of variables which represents one circle on the map should be written in a product format.

2. Take each member of the equation and sum them together. This will generate a simplified SOP equation for this logic function.

The advantage to designing logic circuits with the Karnaugh Map is that all the laws and rules of Boolean Algebra are automatically applied. The disadvantage is that for more than a four variable map, it is difficult to identify adjacent cells easily and for more than a six variable map, computer simplification is strongly advised.

PROCEDURE

1. For below table, obtain the function, F, that has value 1.

|  |  |  |
| --- | --- | --- |
| **Input** | | **Output** |
| **A** | **B** | **F** |
| **0** | **0** | **0** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **1** |

Check the function is

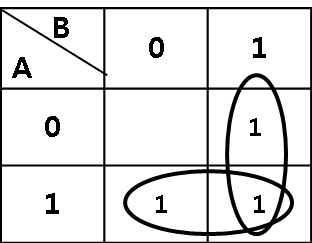
1. Check the circuit of the function is the same as below figure.



1. Fill the below table and check the result is the same as Table in ①.

|  |  |  |
| --- | --- | --- |
| **Input** | | **Output** |
| **A** | **B** | **F** |
| **0** | **0** |  |
| **0** | **1** |  |
| **1** | **0** |  |
| **1** | **1** |  |

1. Using K-map, simplify Table in ①. Combine variables for the function F that has value 1, and check the result is the same as F=A+B, which is shown in below Table. If the result is F=A+B, check the output of the function F is the same as the Table of ③.



|  |  |  |
| --- | --- | --- |
| **V** | | **Output** |
| **A** | **B** | **F** |
| **0** | **0** |  |
| **0** | **1** |  |
| **1** | **0** |  |
| **1** | **1** |  |

1. For below table, obtain the function, F, that has value 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Decimal** | **Input** | | | | **Output** |
| **i** | **A** | **B** | **C** | **D** | **F** |
| **0** | **0** | **0** | **0** | **0** | **0** |
| **1** | **0** | **0** | **0** | **1** | **0** |
| **2** | **0** | **0** | **1** | **0** | **1** |
| **3** | **0** | **0** | **1** | **1** | **1** |
| **4** | **0** | **1** | **0** | **0** | **0** |
| **5** | **0** | **1** | **0** | **1** | **0** |
| **6** | **0** | **1** | **1** | **0** | **1** |
| **7** | **0** | **1** | **1** | **1** | **1** |
| **8** | **1** | **0** | **0** | **0** | **0** |
| **9** | **1** | **0** | **0** | **1** | **1** |
| **10** | **1** | **0** | **1** | **0** | **0** |
| **11** | **1** | **0** | **1** | **1** | **1** |
| **12** | **1** | **1** | **0** | **0** | **0** |
| **13** | **1** | **1** | **0** | **1** | **0** |
| **14** | **1** | **1** | **1** | **0** | **0** |
| **15** | **1** | **1** | **1** | **1** | **1** |

Check the F is the same as below

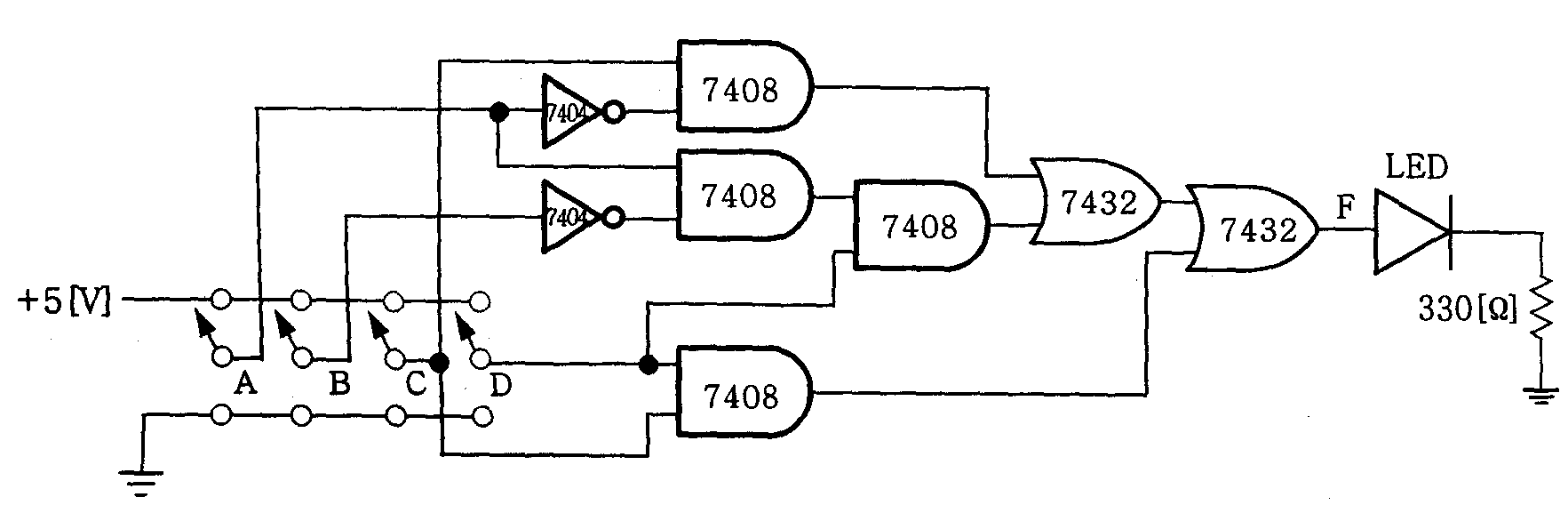
1. Using K-map, obtain simplified F.

Check your answer with below Table and a function F



1. Build logic gates with your answer and check the output result. Compare your result with the Table of ⑤. Your result should be the same as in ⑤. If not, your simplification and logic circuits are wrong.

This logic circuit is drown from equation in ⑥.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Decimal** | **Input** | | | | **Output** |
| **i** | **A** | **B** | **C** | **D** | **F** |
| **0** | **0** | **0** | **0** | **0** |  |
| **1** | **0** | **0** | **0** | **1** |  |
| **2** | **0** | **0** | **1** | **0** |  |
| **3** | **0** | **0** | **1** | **1** |  |
| **4** | **0** | **1** | **0** | **0** |  |
| **5** | **0** | **1** | **0** | **1** |  |
| **6** | **0** | **1** | **1** | **0** |  |
| **7** | **0** | **1** | **1** | **1** |  |
| **8** | **1** | **0** | **0** | **0** |  |
| **9** | **1** | **0** | **0** | **1** |  |
| **10** | **1** | **0** | **1** | **0** |  |
| **11** | **1** | **0** | **1** | **1** |  |
| **12** | **1** | **1** | **0** | **0** |  |
| **13** | **1** | **1** | **0** | **1** |  |
| **14** | **1** | **1** | **1** | **0** |  |
| **15** | **1** | **1** | **1** | **1** |  |

1. Design an invalid BCD Code detector. BCD is a 4-bit binary code representing the decimal numbers 0 through 9. The binary numbers 1010 through 1111 are not used in BCD.
2. Construct a truth table containing all possible inputs and desired output. Assume that the desired output for a valid code is a 1, and for an invalid code is 0. Complete the truth table below. A is the most significant bit, and D is the least significant bit.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **K-Map** | | | | |
| **A** | **B** | **C** | **D** | **Y** |
| **0** | **0** | **0** | **0** |  |
| **0** | **0** | **0** | **1** |  |
| **0** | **0** | **1** | **0** |  |
| **0** | **0** | **1** | **1** |  |
| **0** | **1** | **0** | **0** |  |
| **0** | **1** | **0** | **1** |  |
| **0** | **1** | **1** | **0** |  |
| **0** | **1** | **1** | **1** |  |
| **1** | **0** | **0** | **0** |  |
| **1** | **0** | **0** | **1** |  |
| **1** | **0** | **1** | **0** |  |
| **1** | **0** | **1** | **1** |  |
| **1** | **1** | **0** | **0** |  |
| **1** | **1** | **0** | **1** |  |
| **1** | **1** | **1** | **0** |  |
| **1** | **1** | **1** | **1** |  |

1. Draw the Karnaugh map, and write the simplified Boolean expression for the valid codes as sum of products.
2. Draw the circuit for the above simplified Boolean expression. And build the circuit and check the result with a).