

## Lab 4: Combinational Logic Design

### A. Objectives

- Design a complete minimal combinational logic system from specification to implementation.
- Minimize combinational logic circuits using Karnaugh maps.
- Learn various numerical representation systems.
- Implement circuits using canonical minimal forms.

### B. Theory

Computers can accept only binary values whereas we usually need to work with decimal numbers. Therefore, we must represent the decimal digits by means of a code that contains 1's and 0's in order for computers to understand the values and perform operations using them. The code most commonly used for the decimal digits is the straight binary assignment. This scheme is called binary-coded decimal and is commonly referred to as BCD. A decimal number in BCD is the same as its equivalent binary number only when the number is between 0 and 9.

Excess-3 is another decimal code in which each coded combination is obtained from the corresponding binary value plus 3.

Gate-level minimization is the design task of finding an optimal gate-level implementation of the Boolean functions describing a digital circuit. However, the procedure of minimization is awkward because it lacks specific rules to predict each succeeding step in the manipulative process. The map method, on the other hand, provides a simple, straightforward procedure for minimizing Boolean functions. This method may be regarded as a pictorial form of a truth table. The map method is also known as the Karnaugh map or K-map.

A K-map is a diagram made up of squares, with each square representing one minterm of the function that is to be minimized. In fact, the map presents a visual diagram of all possible ways a function may be expressed in standard form. By recognizing various patterns, it is possible to derive alternative algebraic expressions for the same function, from which the simplest can be selected.

**Figure B1** shows the minterm positions on the K-map for 4 input variables and 1 output variable.

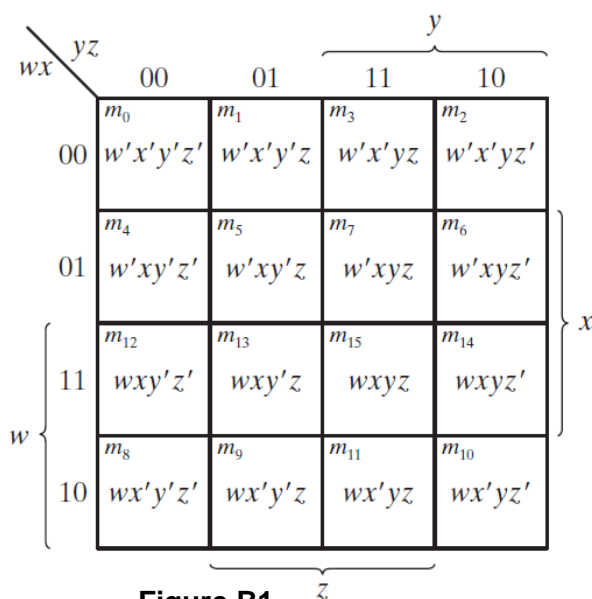
#### Example

**Table B1** shows the truth table for the function  $A = WX + XYZ' + WZ + X'Y'$  where WXYZ goes from binary 0 to 9.

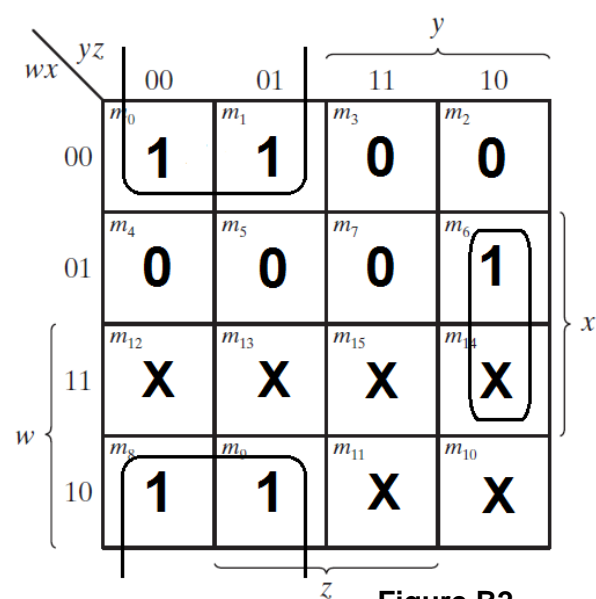
**Figure B2** shows the K-map for the function. We can use the K-map to minimize the function to  $A = X'Y' + XYZ'$

| W | X | Y | Z | A |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |

**Table B1**



**Figure B1**



**Figure B2**

### C. Apparatus

- Trainer Board
- 1 x IC 4073 Triple 3-input AND gates
- 1 x IC 4075 Triple 3-input OR gates
- 1 x IC 7404 Hex Inverters (NOT gates)
- 1 x IC 7400 2-input NAND gates
- 2 x IC 7408 2-input AND gates

### D. Procedure

Design of a BCD to Excess-3 converter: Design, minimize and implement a digital logic system where an input in binary coded decimal (BCD) is converted and displayed in Excess-3.

1. Complete the truth table (**Table F1**, Section F) for the BCD to Excess-3 converter.
2. Identify the inputs and outputs from the truth table and complete the system analysis (**Table F2**, Section F).
3. Complete the K-maps (**Figure F1**, Section F) to find the minimal 1st canonical functions of each output variable.
4. Draw the minimal circuit showing the pin configurations (**Figure F2**, Section F).
5. Implement and test the circuit on the trainer board. Test one output at a time.
6. Connect the 4 inputs to the BCD inputs on the trainer board to display the input digits on the seven-segment display.
7. Convert, implement and test the circuit in the suitable universal gate format. Show the circuit with pin configurations (**Figure F3**, Section F).

### E. Report

1. Draw the IC diagram for output **B** of the BCD to Excess-3 converter.
2. Simulate the entire BCD to Excess-3 converter in Logisim. Provide a screenshot of the Logisim circuit schematic and truth table with your report.

**F. Experimental Data**

| Decimal Digit | Binary Coded Decimal (BCD) |   |   |   | Excess-3 |   |   |   |
|---------------|----------------------------|---|---|---|----------|---|---|---|
|               | W                          | X | Y | Z | A        | B | C | D |
| 0             |                            |   |   |   |          |   |   |   |
| 1             |                            |   |   |   |          |   |   |   |
| 2             |                            |   |   |   |          |   |   |   |
| 3             |                            |   |   |   |          |   |   |   |
| 4             |                            |   |   |   |          |   |   |   |
| 5             |                            |   |   |   |          |   |   |   |
| 6             |                            |   |   |   |          |   |   |   |
| 7             |                            |   |   |   |          |   |   |   |
| 8             |                            |   |   |   |          |   |   |   |
| 9             |                            |   |   |   |          |   |   |   |

**Table F1: Truth table - BCD to Excess-3**

|                                |  |                          |  |
|--------------------------------|--|--------------------------|--|
| <b>Number of inputs bits:</b>  |  | <b>Input variables:</b>  |  |
| <b>Number of outputs bits:</b> |  | <b>Output variables:</b> |  |

**Table F2: System analysis**

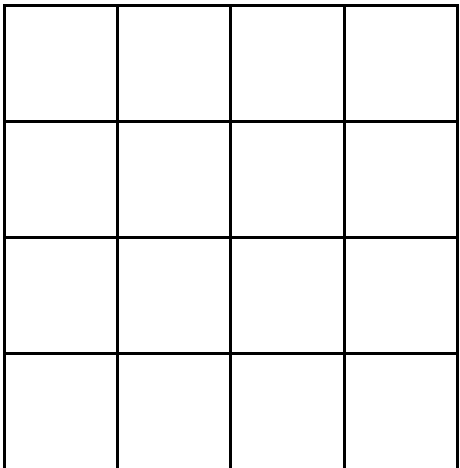
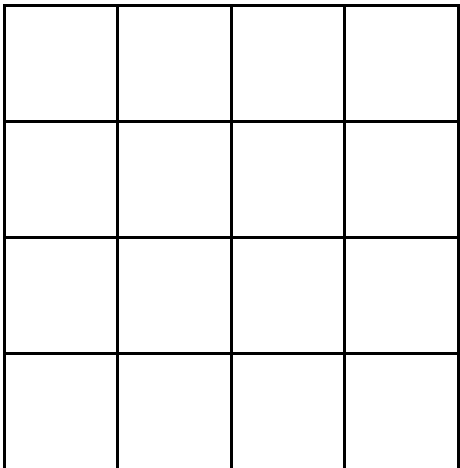
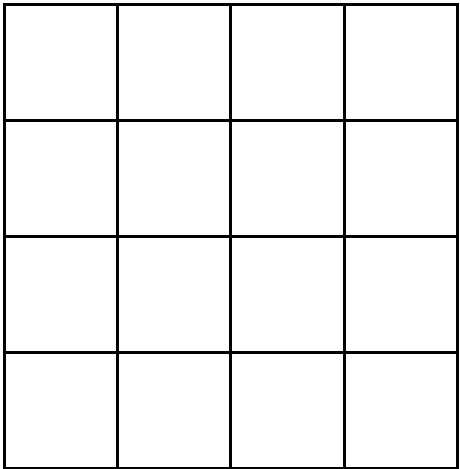
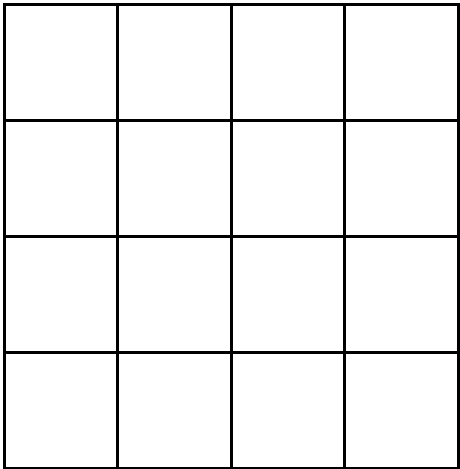


Figure F1: K-Maps

**Figure F2: Minimal 1st canonical circuit of BCD to Excess-3 converter**

**Figure F3: Minimal universal gate implementation of BCD to Excess-3 converter**