

**CMPE 160 Laboratory Exercise 5**  
**Combinational Logic Circuit Design Using Karnaugh Map**  
**Simplification**

Andrei Tumbar  
Performed: February 13th  
Submitted: February 20th

Lab Section: 4  
Instructor: Mr. Byers  
TA: Sam Myers  
Kobe Balin  
Georgi Thomas

Lecture Section: 1  
Professor: Mr. Cliver

By submitting this report, you attest that you neither have given nor have received any assistance (including writing, collecting data, plotting figures, tables or graphs, or using previous student reports as a reference), and you further acknowledge that giving or receiving such assistance will result in a failing grade for this course.

Your Signature: \_\_\_\_\_

# Abstract

In this laboratory exercise, an arbitrary function was implemented using a simplified Karnaugh map. A product of sums and a sums of product expression was generated. Both circuits were designed in Quartus II and then simulated in Modelsim. A forces file was initially used to control input pins, however, the test bench provided on MyCourses proved to be a more efficient way to test the circuit. The cost of each circuit was evaluated. The cheap circuit was implemented a breadboard. The function was a four-input, one-output function described using min-term notation. The exercise was successful when implementing the circuit on a breadboard as well in the Modelsim simulation.

## Design Methodology

A function provided in equation 1 is a min-term representation of  $F$ .

$$F = \sum_{ABCD}(0, 2, 3, 4, 6, 7, 13, 14, 15) \quad (1)$$

Every number above shows in the input combinations that would result in a 1 in binary. The 4-bit input is represented by  $A$ ,  $B$ ,  $C$ , and  $D$  in that order.

## Karnaugh Maps

Two Karnaugh Maps were created to implement the function using max-term and min-terms. The sum-of-products (min-terms) and product-of-sums (max-terms) expressions were found below.

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

(a) Min-terms K-Map

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

(b) Max-terms K-Map

Figure 1: K-Maps for both POS and SOP functions

The groups in Figure 1 were written as a sum-of-products (SOP) and a product-of-sums (POS). Figure 1a shows 4 groupings and therefore 4 terms were generated. Figure 1b however, shows 3 groupings.

An expression can be extracted from each expression. This results in a boolean expression in POS or SOP form.

$$F_{POS} = (\bar{A} + B)(\bar{A} + C + D)(A + C + \bar{D}) \quad (3)$$

Figure 2 is a circuit diagram for both the POS and SOP implementations of  $F$ . It displays the pin numbers to the chips in the format `CHIP_PIN`. Both input and output pins are shown on either side of every node. When only one chip of that type is used in the function, a chip

number is not included. For example, only one AND chip is needed and therefore the labels just display the pin number on the single chip.

## Results and Analysis

The Quartus II design was tested using the VHDL test bench as signal inputs. A wave form for the POS and SOP functions was generated.

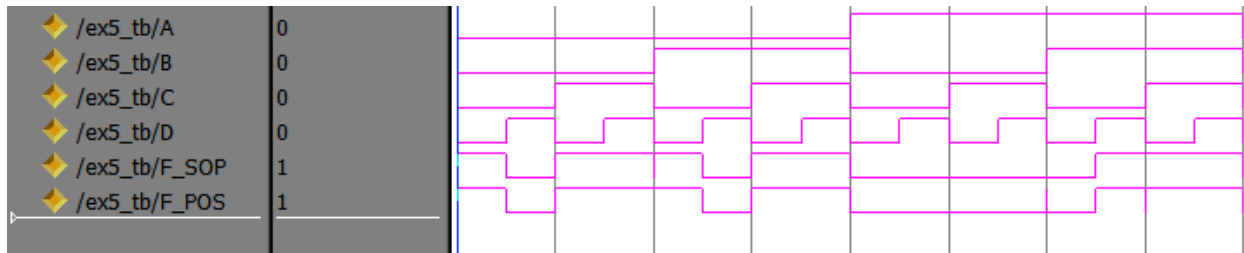


Figure 3: Circuit simulation using the test bench

Figure 3 shows all of the input combinations of the 4-input function. The two outputs generate the same waveform because they are different implementations of the same function.

## Conclusion

The purpose of this exercise was to analyze the efficiency of min-term (SOP) and max-term (POS) function implementation. An arbitrary function was provided. Two Karnaugh maps were generated and used to extract the SOP and POS expressions. These expressions were then implemented in a circuit in Quartus II and simulated in Modelsim. The cheaper circuit was then constructed on a breadboard. This experiment was successful in obtaining the correct output set as provided by the function from the breadboard and the simulation.

# Questions

1. Design a two level NAND-NAND implementation of  $F_{SOP}$ .

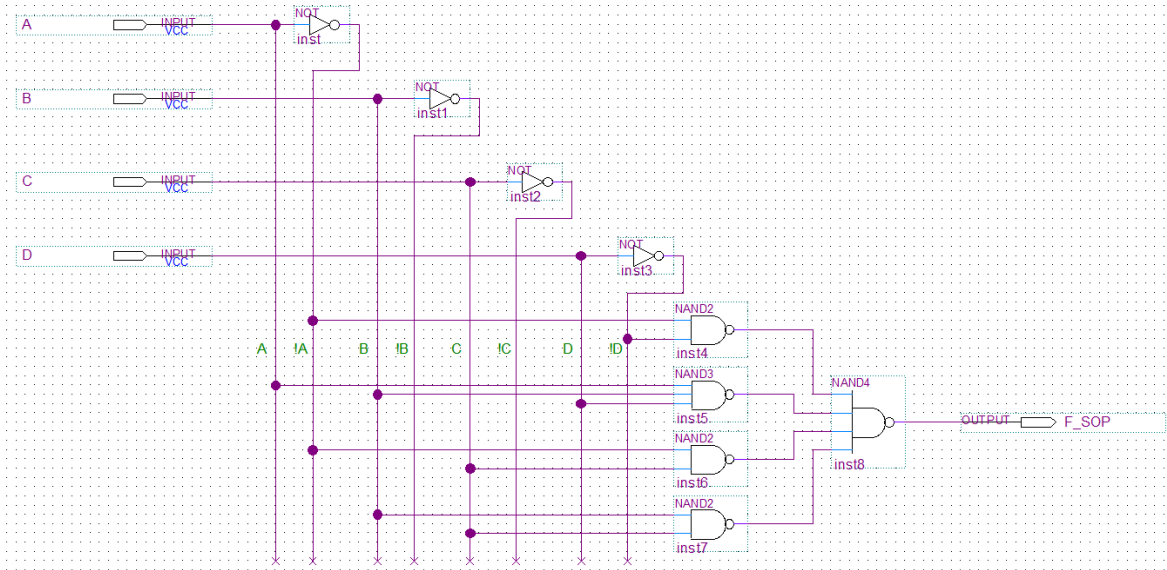


Figure 4: Two level NAND-NAND implementation of  $F_{SOP}$

2. Design a two level NOR-NOR implementation of  $F_{POS}$ .

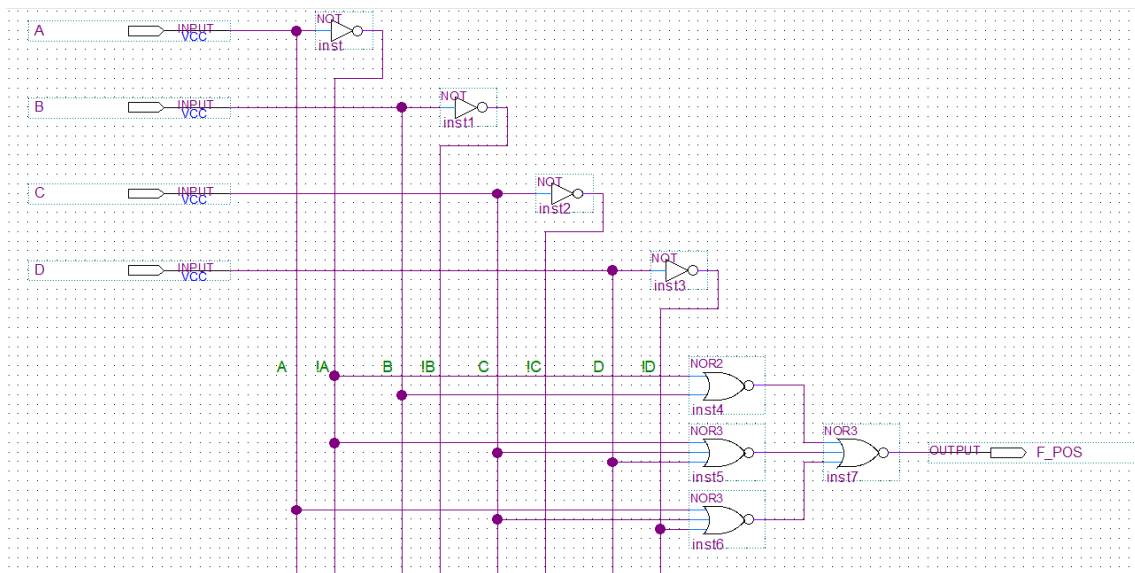


Figure 5: Two level NOR-NOR implementation of  $F_{POS}$