Post Lab Report-02

Date: 27 March 2021	Course: CSE345 (Digital Logic Design)
Experiment 2	Id: 2019-1-60-024
Name: Adri Saha	Course instructor: Touhid Ahmed

Experiment Name: Design and Implementation of a Combinational Circuit

Abstract: One of the introductory courses for Electrical Engineering and Electrical Engineering Technology students is logic circuit design. It can introduce students to circuit design, problem solving, testing, and feature verification. Students were asked to design some combinational logic circuits by using Karnaugh map. And they can also implement combinational circuit using AND-OR and OR-AND logic from using the K map. Here, they also find Sum of product (SOP) and Product of Sum (POS) function expressions from Karnaugh map and simplify the function. For each logic circuit, they can also learn how connect four inputs to four data switches and the output to a LED indicator. And that is how they can design a complete logic circuits and also can get to know about the uses of these. The digital logic design lab is a learning experience that most students enjoy because it gives them their first hands-on experience designing and constructing miniature structures.

Introduction:

Combinational Logic Circuits: Digital logic circuits may be combinational or sequential. A combinational circuit consists of input variables, logic gates, and output variables. The logic gates accept signals from the input and generate signals to the outputs. The principle of operation is that the circuit operates on just two voltage levels, called logic 0(low) and logic 1(high).

For n input variables, there are 2^n possible combinations of binary input values.

Uses:

- Use in computer circuits to perform Boolean algebra on input signals and on stored data.
- Practical computer circuits normally contain a mixture of combinational and sequential logic.

Objective: Our main goal is to design logic circuits which produce a high (logic 1) output only when the binary number is greater than 0101_2 which means from index 6 the voltage will be high. Here truth tables, Karnaugh map, SOP, POS, IC's, LED, Schematic and Structural Verilog Simulation of Combinational Logic Circuits are required to do this experiment.

- Creating a project using Quartus II software.
- Design entry using Verilog code and schematic files.
- Assigning the circuit inputs and finding specific outputs.
- Simulating the designed circuit.
- Minimize Boolean expression using k-map.
- Design a combinational circuit from descriptive problem specification.
- Implement a combinational circuit using AND-OR and OR-AND logic.

Theory and experiments results:

A. **<u>Karnaugh Map:</u>** The Karnaugh Map (in short K-map) is a special pictorial form of a truth table. This map provides a simple straightforward procedure for representation and simplification of Boolean functions. It gives simplified form to a function. So, the K-map method of solving the logical expressions is referred to as the graphical technique of simplifying Boolean expressions.

We can Simplify of Boolean expressions using Karnaugh Map by

- Define the given expression in its canonical form.
- Create the K-map by entering 1 to each product-term into the K-map cell and fill the remaining cells with zeros.
- Form the groups by considering each one in the K-map.
- Each group should have the largest number of 'ones'. A group cannot contain an empty cell or cell that contains 0
- In a group, there is a total of 2^n number of ones. Here, n=0, 1, 2, …n.

- **Example:** 2^0=1, 2^1=2, 2^2=4, 2^3=8, or 2^4=16.

Uses:

- Truth tables are translated into Karnaugh maps, which offer a visual representation of a much simpler formula for expressing the same logic. However, once anyone construct a truth table, he/she can and should create a K-Map to simplify the logic.
- Karnaugh maps are used to simplify real-world logic specifications so they can be implemented with the minimum possible logic gates.
- The Quine–McCluskey method is a tabular method that outperforms Karnaugh maps when there are a lot of inputs.

B. **SOP & POS:** The methods for deducing a particular logic function are SOP (Sum of Product) and POS (Product of Sum). In other words, these are the different ways to represent the reduced logic function that has been deduced. The deduced logic function can be used to build a logic circuit.

1. **Sum of Product (SOP):** In Karnaugh Map, when we find simplified function of SOP we take the high voltage or output 1 from K-map. For SOP we put 1's in blocks of K-map respective to the minterms (0's elsewhere).

2. **Product of Sum (POS):** In Karnaugh Map, when we find simplified function of SOP we take the low voltage or output 0 from K-map. For POS we put 0's in blocks of **K-map** respective to the maxterms(1's elsewhere)

Uses:

- The methods for deducing a particular logic function are SOP (Sum of Product) and POS (Product of Sum). In other words, these are the different ways to represent the reduced logic function that has been deduced.
- The deduced logic function can be used to build a logic circuit.

C. **AND-OR Logic gates:** From given problem’s K-map for SOP, we get AND-OR logic expression for function of Sum of Product. The expression is here: A+BC. Here, B & C inputs relate to and gate. And then B C & A relate to OR logic gate in the output.

D. **OR-AND logic gates:** On the other hand, from given problem's K-map for POS, we get OR-AND logic expression for function of Product of Sum. The expression is here: (A+B) (A+C). Here, A & B inputs relate to OR gate. And then B & C relate to OR logic gate. Then (A+B) & (B+C) are related to and gate to the output. So, here it is the OR-AND logic expression.

Truth Table:

Here, inputs are A, B, C, D(4 bit binary number) and output is F where A is the MSB and D is the LSB>The output is high (logic 1) output only when the binary number is greater than 01012.

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
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	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

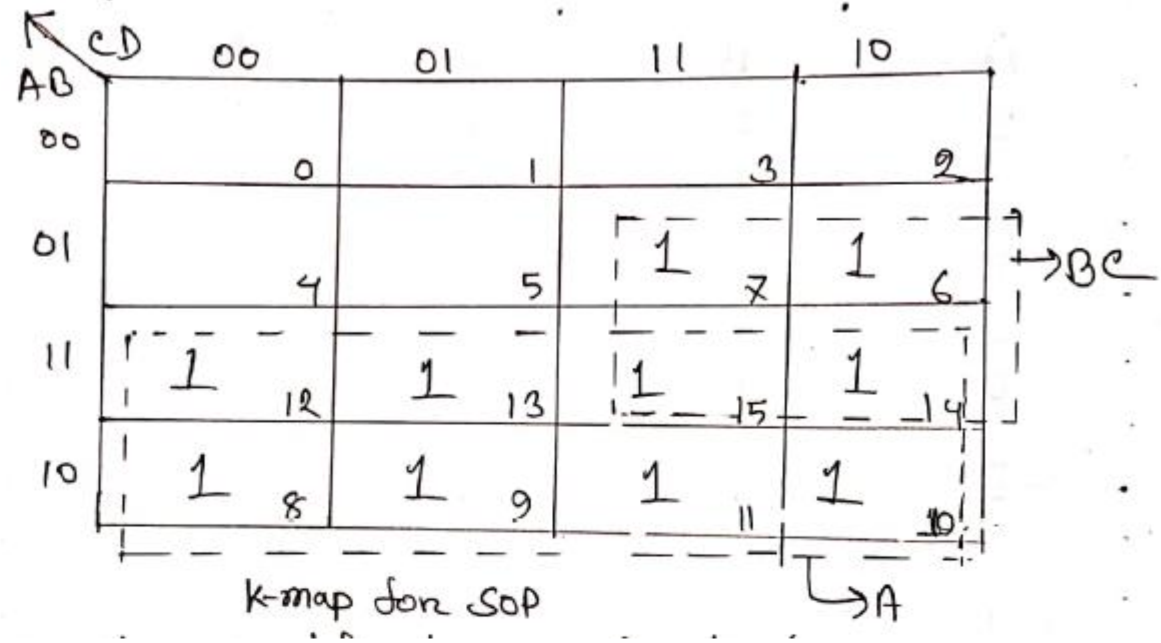
Figure 1: Truth Table for descriptive Problem

A. Simplified Sum of Product (POS) from Karnaugh map:

Now, the canonical form of sum of product (SOP) can be written as,

$F_1(A, B, C, D) = \sum m\ (6,7,8,9,10,11,12,13,14,15)$

Karnaugh Map as follows:



AB \ CD	00	01	11	10
00	0	1	3	2
01	4	5	1 7	1 6
11	1 12	1 13	1 15	1 14
10	1 8	1 9	1 11	1 10

Figure 2: K-map for SOP

1st Boolean expression is:

$$F_1 = A+BC$$

Schematic Circuit:

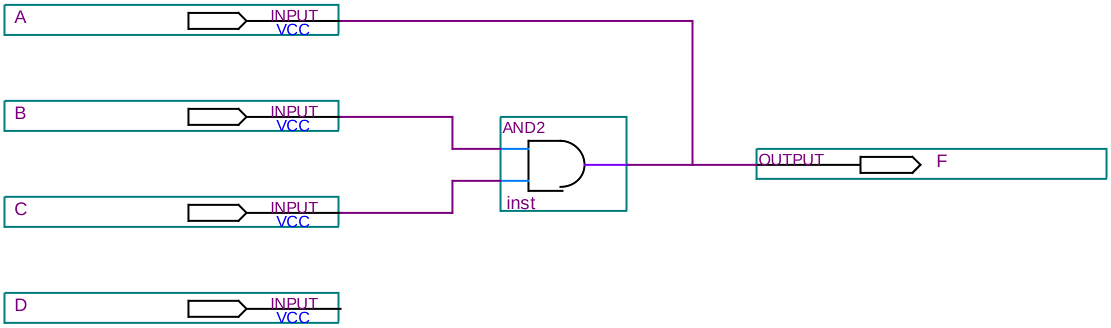


Figure 3 : Logic Circuit for Simplified SOP

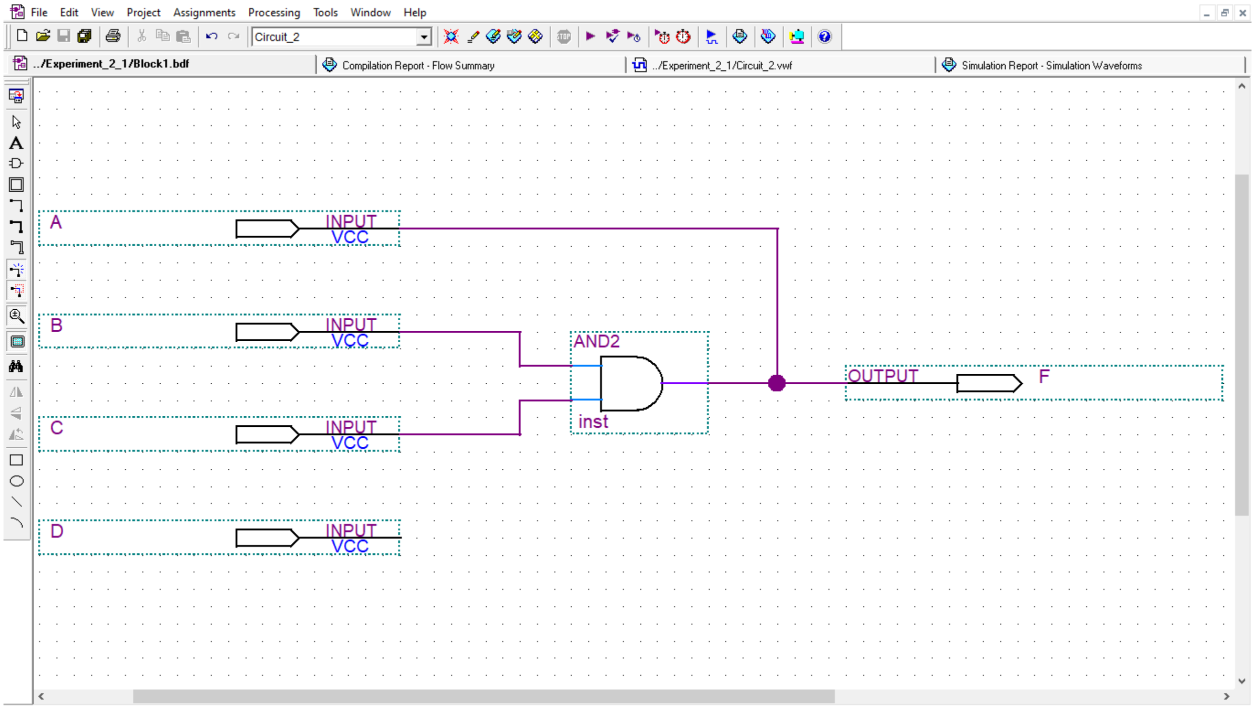

File
Edit
View
Project
Assignments
Processing
Tools
Window
Help
Circuit_2
../Experiment_2_1/Block1.bdf
Compilation Report - Flow Summary
../Experiment_2_1/Circuit_2.vwf
Simulation Report - Simulation Waveforms
A
INPUT
VCC
B
INPUT
VCC
C
INPUT
VCC
D
INPUT
VCC
AND2
inst
OUTPUT
F

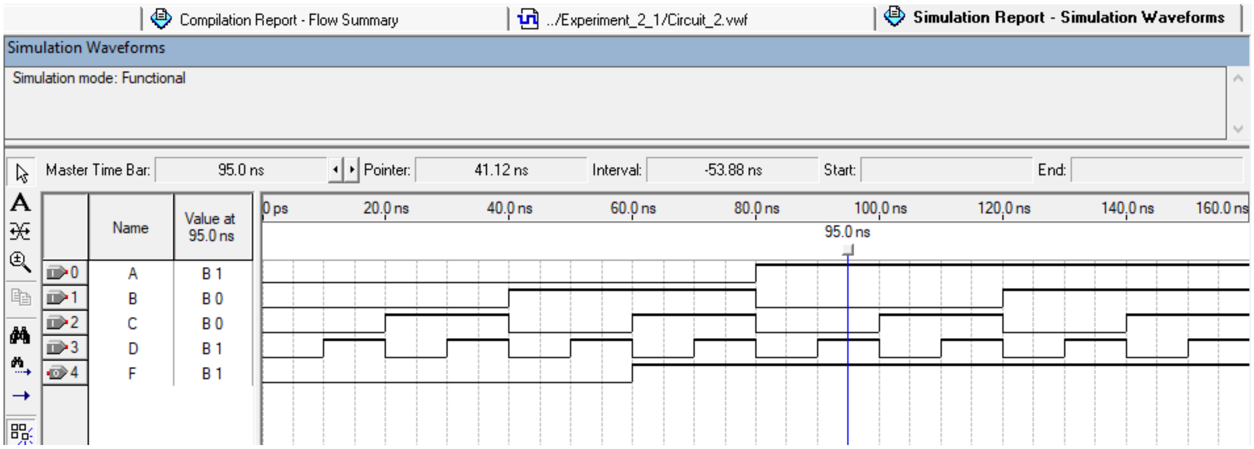
Figure 4: Logic Circuit for Simplified SOP Screenshot

Schematic drawing: This is a schematic of the design using primitive components such as logic gates, flipflops, decoders, encoders and wires for interconnections. There is a list of components in the software library that users can pick from.

Schematic Waveform Simulation:

In this simulation, we have taken end time 160 nano second for 4 input A,B,C,D. So, time period of A,B,C & D is 160, 80, 40 & 20 nano seconds in sequent. As there are 4 inputs.

We see, output is “high” or 1 after 60 nano seconds.


Compilation Report - Flow Summary
../Experiment_2_1/Circuit_2.vwf
Simulation Report - Simulation Waveforms
Simulation Waveforms
0 ps
20.0 ns
40.0 ns
60.0 ns
80.0 ns
100.0 ns
120.0 ns
140.0 ns
160.0 ns
95.0 ns

Simulation mode: Functional

Master Time Bar: 95.0 ns Pointer: 41.12 ns Interval: -53.88 ns Start: End:

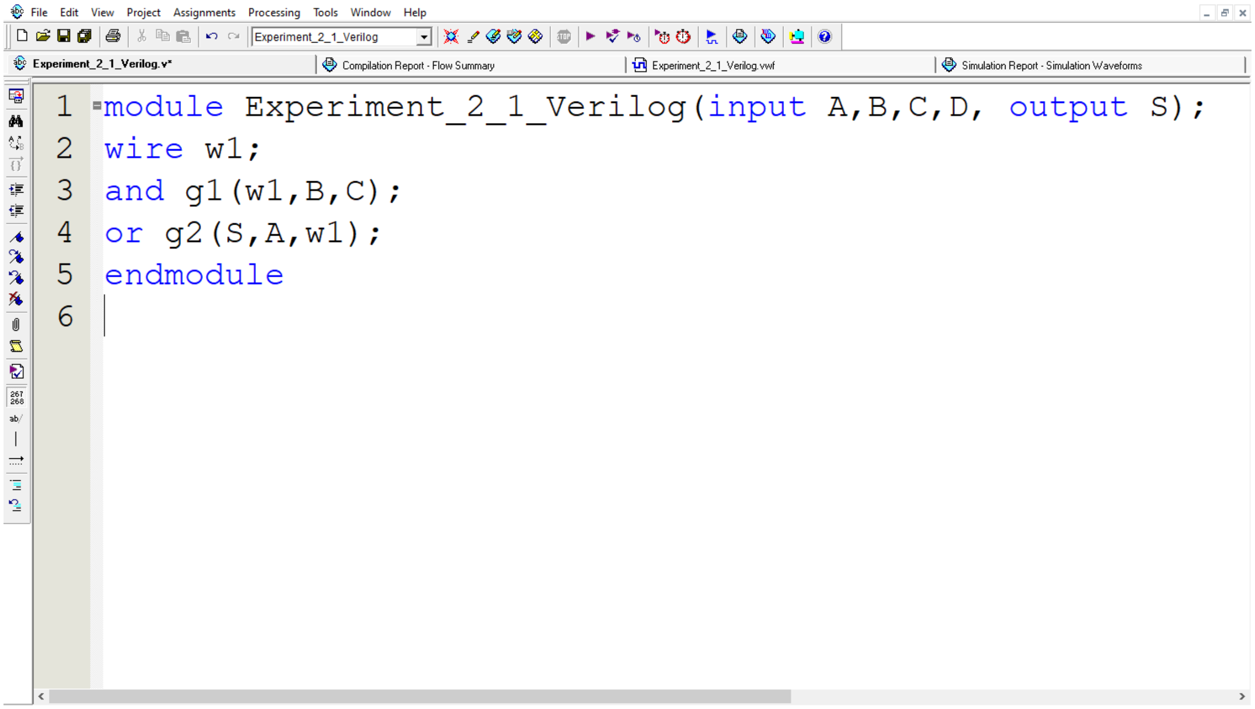
	Name	Value at 95.0 ns
0	A	B 1
1	B	B 0
2	C	B 0
3	D	B 1
4	F	B 1

Figure 5: Simulated Waveform by simplified SOP logic circuit

Verilog Codes: Verilog is a Hardware Description Language; a textual format for describing electronic circuits and systems.

Uses:

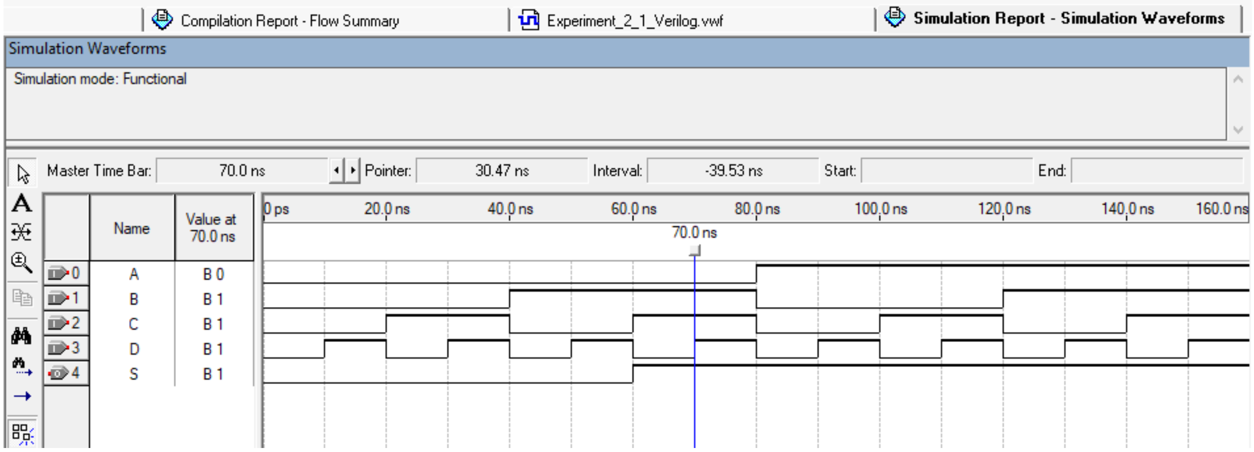
- Applied to electronic design.
- Verilog is intended to be used for verification through simulation, for timing analysis, for test analysis (testability analysis and fault grading) and for logic synthesis.


File Edit View Project Assignments Processing Tools Window Help
Experiment_2_1_Verilog
Experiment_2_1_Verilog.v*
Compilation Report - Flow Summary
Experiment_2_1_Verilog.vwf
Simulation Report - Simulation Waveforms

```
module Experiment_2_1_Verilog(input A,B,C,D, output S);
wire w1;
and g1(w1,B,C);
or g2(S,A,w1);
endmodule
```

Figure 6: Verilog Codes for Simplified SOP

Verilog simulation:



Compilation Report - Flow Summary | Experiment_2_1_Verilog.vwf | Simulation Report - Simulation Waveforms

Simulation Waveforms

Simulation mode: Functional

Master Time Bar: 70.0 ns | Pointer: 30.47 ns | Interval: -39.53 ns | Start: | End:

	Name	Value at 70.0 ns
0	A	B 0
1	B	B 1
2	C	B 1
3	D	B 1
4	S	B 1

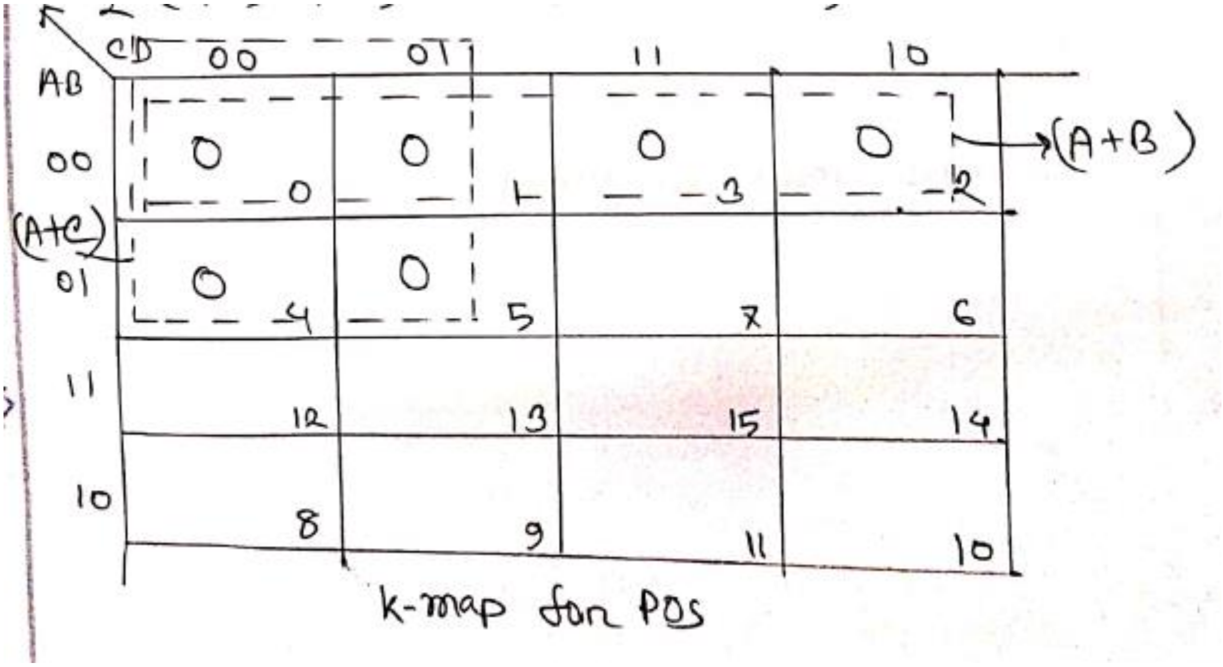
Figure 7: Verilog Codes simulation for Simplified SOP

B. <u>Simplified Product of Sum(SOP) from Karnaugh map:</u> The canonical form of product of sum can be written as,

2nd expression:

$$\mathbf{F_2(A, B, C, D) = \sum M\ (0,1,2,3,4,5)}$$

The Karnaugh Map is as follows:



AB \ CD	00	01	11	10
00	0 0	0 1	0 3	0 2
01	0 4	0 5	7	6
11	12	13	15	14
10	8	9	11	10

k-map for POS

Figure 8: K-map for POS

The circuit diagram:

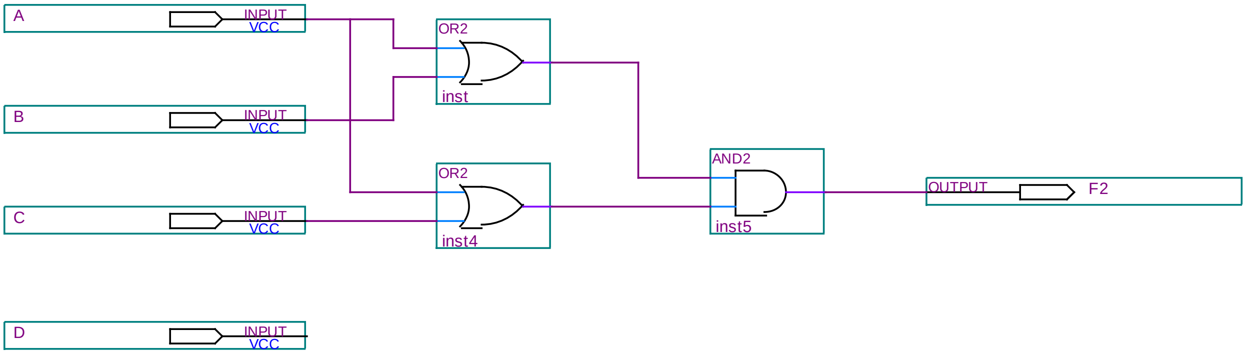


Figure 9: Logic Circuit for simplified Product of Sum function

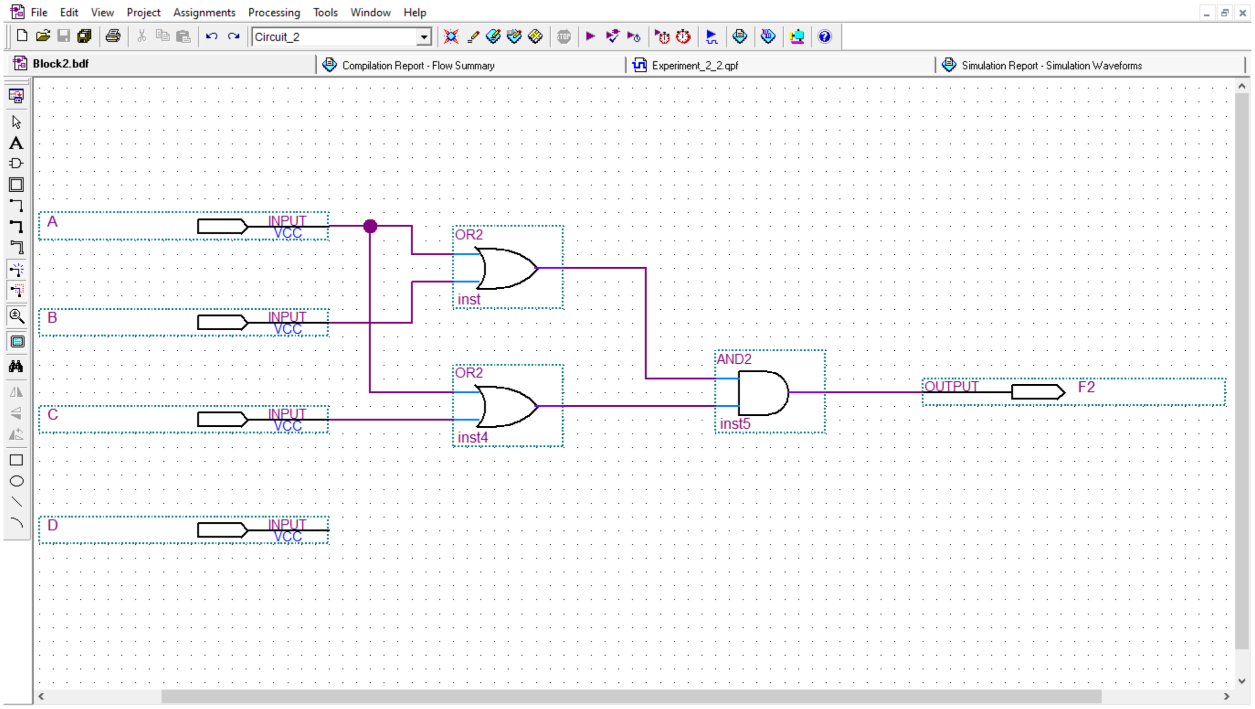
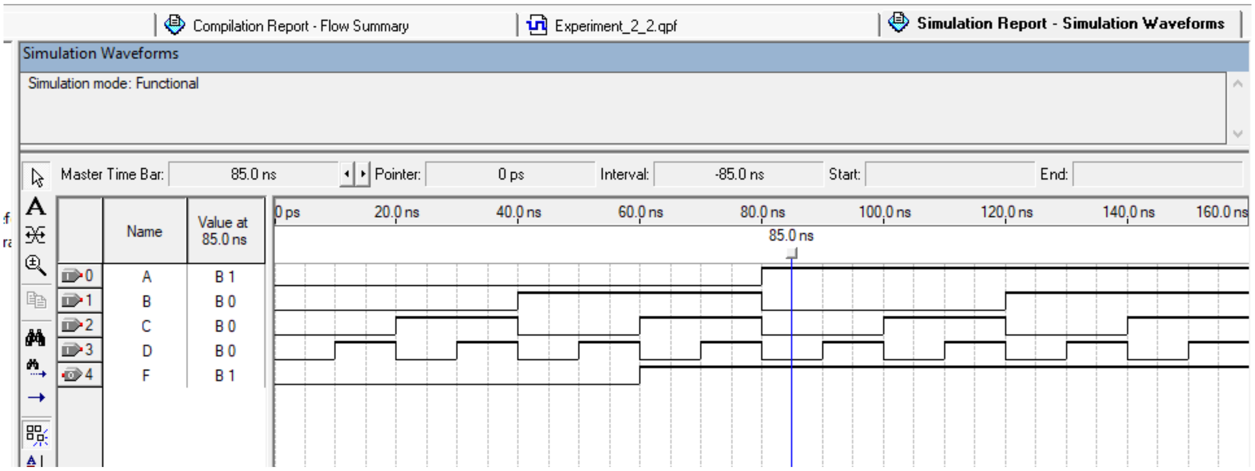

File Edit View Project Assignments Processing Tools Window Help
Circuit_2
Block2.bdf
Compilation Report - Flow Summary
Experiment_2_2.qpf
Simulation Report - Simulation Waveforms
A
INPUT
VCC
B
INPUT
VCC
C
INPUT
VCC
D
INPUT
VCC
OR2
inst
OR2
inst4
AND2
inst5
OUTPUT
F2

Figure 10: Logic Circuit for simplified Product of Sum function Screenshot

Simulation: In this simulation, we have taken end time 160 nano second for 4 input A, B, C, D. So, time period of A, B, C & D is 160, 80, 40 & 20 nano seconds in sequent. As there are 4 inputs.

We see, output is “high” or 1 after 60 nano seconds.



Simulation Waveforms

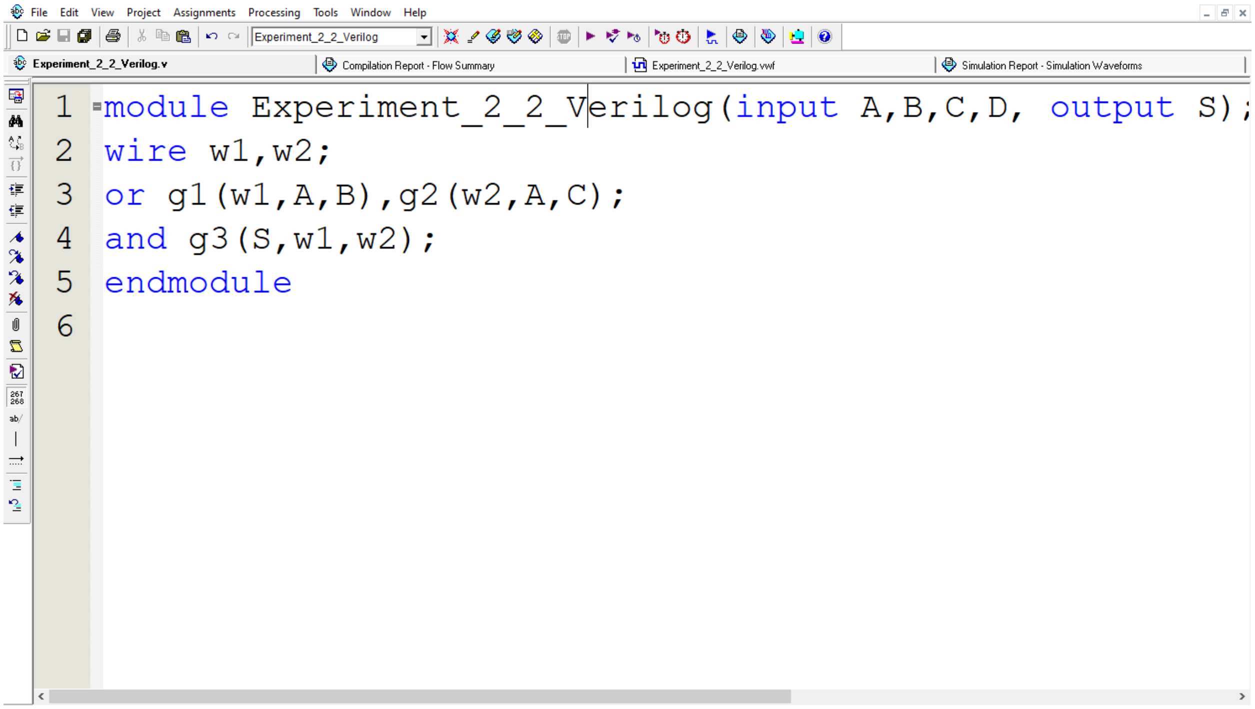
Simulation mode: Functional

Master Time Bar: 85.0 ns Pointer: 0 ps Interval: -85.0 ns Start: End:

	Name	Value at 85.0 ns
0	A	B 1
1	B	B 0
2	C	B 0
3	D	B 0
4	F	B 1

Figure 11: Simulated Waveform by simplified POS logic circuit

Verilog Codes:

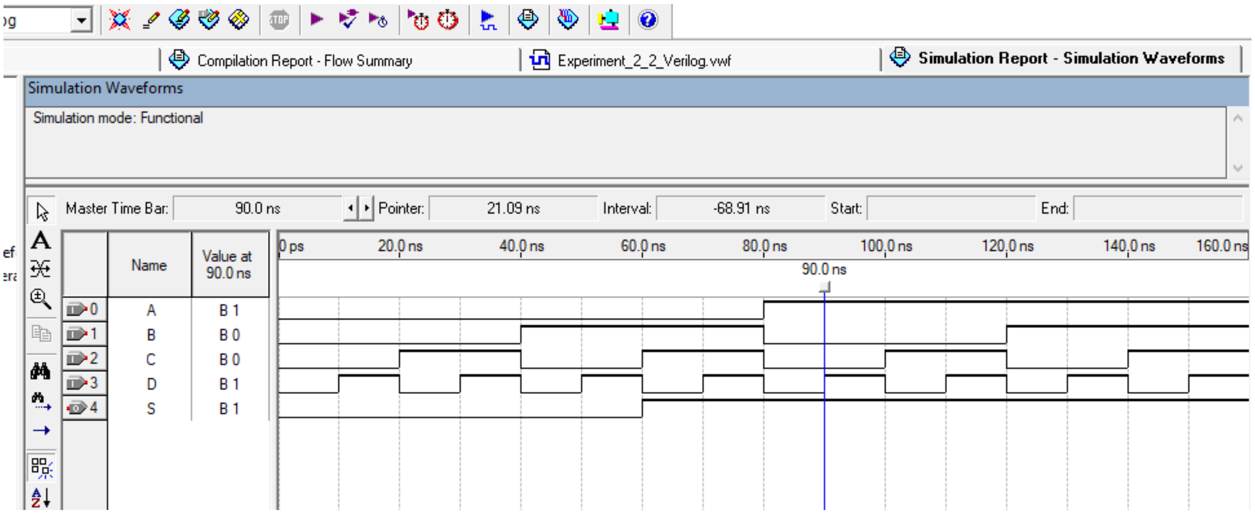


```
module Experiment_2_2_Verilog(input A,B,C,D, output S);
wire w1,w2;
or g1(w1,A,B),g2(w2,A,C);
and g3(S,w1,w2);
endmodule

```

Figure 12: Verilog Codes for Simplified POS

Verilog Simulation:


Compilation Report - Flow Summary
Experiment_2_2_Verilog.vwf
Simulation Report - Simulation Waveforms
0 ps
20.0 ns
40.0 ns
60.0 ns
80.0 ns
100.0 ns
120.0 ns
140.0 ns
160.0 ns
90.0 ns

Simulation Waveforms

Simulation mode: Functional

Master Time Bar: 90.0 ns Pointer: 21.09 ns Interval: -68.91 ns Start: End:

	Name	Value at 90.0 ns
0	A	B 1
1	B	B 0
2	C	B 0
3	D	B 1
4	S	B 1

Figure 13: Verilog Codes simulation for Simplified POS

Conclusion: From this experiment, students learned minimizing a Boolean expression from using Karnaugh maps. They also learn to design a combinational circuit from descriptive problem specification. to verify a Boolean identity such as the distributive property truth tables are also used for Karnaugh map. They build the circuit and test the output for all simplified possible input combinations. They also know about uses of SOP & POS in Karnaugh map. And also verify his/her hands-on experience designing by Quartus Software.

Reference:

- Digital Logic Design; Md Mozammel Huq Azad Khan.
- https://www.javatpoint.com/simplification-of-boolean-expressions-using-karnaugh-map