KIIT, Deemed to be University
School of Electronics Engineering
Digital System Design Laboratory [EC 29005]

EXPERIMENT - 1

Aim:

Design and Simulation of boolean functions using Verilog HDL.

Hardware implementation of a Boolean function in sum of products and product of sums expressions using universal gates.

Component/Software Used:

Component/Software	Specification
ICs	7400, 7402
Bread Board, Power supply, LEDs, Resistors,Switches, Connecting wires	As per requirement
Software(s) Used	Vivado 2016.1

Theory:

A binary variable may appear either in its normal form (A) or in its complement form ($\overline{A}$). Binary variables combined with an AND operation is called **minterms**, or a standard product. Similarly binary variables forming an OR terms called **maxterms**, or standard sums. For **"n"** variables with each variable being primed or unprimed, provide **"2^n"** possible minterms and maxterms.

The eight (2^3) different minterms and maxterms are shown in Table 1.1 for three (3) variables.

A Boolean function can be expressed algebraically from a given truth table by forming a **minterm** for each combination of the variables that produces a **"1"** in the function and then taking the **OR** of all those terms. From a given truth table, a Boolean function can alternatively be written algebraically by constructing a **maxterm** for each combination of variables that yields a **"0"** in the function and then taking the **AND** of all those terms. Boolean functions expressed as a sum of minterms or product of maxterms are said to be in canonical form.The two canonical forms are basic forms of expressing Boolean function which is obtained from reading a given function from the truth table.

			Minterms		Maxterms	
A	B	C	Term	Designation	Term	Designation
0	0	0	$\overline{A}\,\overline{B}\,\overline{C}$	m_0	$A + B + C$	M_0
0	0	1	$\overline{A}\,\overline{B}\,C$	m_1	$A + B + \overline{C}$	M_1
0	1	0	$\overline{A}\,B\overline{C}$	m_2	$A + \overline{B} + C$	M_2
0	1	1	$\overline{A}\,B\,C$	m_3	$A + \overline{B} + \overline{C}$	M_3
1	0	0	$A\,\overline{B}\,\overline{C}$	m_4	$\overline{A} + B + C$	M_4
1	0	1	$A\,\overline{B}\,C$	m_5	$\overline{A} + B + \overline{C}$	M_5
1	1	0	$A\,B\,\overline{C}$	m_6	$\overline{A} + \overline{B} + C$	M_6
1	1	1	$A\,B\,C$	m_7	$\overline{A} + \overline{B} + \overline{C}$	M_7

Table 1.1: Minterms and Maxterms for 3 binary variables

Another way to express Boolean functions is in standard form. In digital logic, the "sum of products" (SOP) and "product of sums" (POS) are two distinct standard forms to describe a Boolean expression. These representations are used to simplify and analyze logic circuits.

Sum of Products (SOP): SOP is a Boolean expression containing AND terms, called product terms, with one or more literals each. The sum denotes the ORing of these terms.

Product of Sums (POS): POS is a Boolean expression containing OR terms, called sum terms. Each term may have any number of literals. The product denotes the ANDing of these terms.

Both SOP and POS representations can be useful in different scenarios, depending on the complexity of the logic circuit and the requirements of the design. In some cases, SOP might be more suitable, while in others, POS might provide a simpler and more concise expression. The choice of representation can affect the number of logic gates required and the overall complexity of the circuit.

Consider a Boolean function $\mathbf{F(A, B, C) = AB\overline{C} + A\overline{B}C + BC}$ for which the minterms and maxterms can be derived from the truth table shown in Table 1.2. This function can be implemented with NAND gates as shown in Figure 1.1. by considering the SOP expression. Also the given function can be implemented with NOR gates as shown in Figure 1.2. by considering the POS expression.

NAND or NOR gates are called Universal gates. Digital circuits are frequently constructed with NAND or NOR gates rather than with AND and OR gates. Universal gates are easier to fabricate with electronic components.

Inputs			Output
A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$$\text{SOP: } F(A,B,C) = m_1 + m_3 + m_6 + m_7$$

$$= \sum (1,3,6,7) = AB + \bar{A}C$$

$$= \overline{\left(\overline{(AB)}\,\overline{(\bar{A}C)}\right)}$$

$$\text{POS: } F(A,B,C) = M_0\, M_2\, M_4\, M_5$$

$$= \prod (0,2,4,5) = (A + C)(\bar{A} + B)$$

$$= \overline{\left(\overline{(A + C)} + \overline{(\bar{A} + B)}\right)}$$

Table 1.2: Truth table of the Boolean function
$F(A, B, C) = AB\bar{C} + A\bar{B}C + BC$

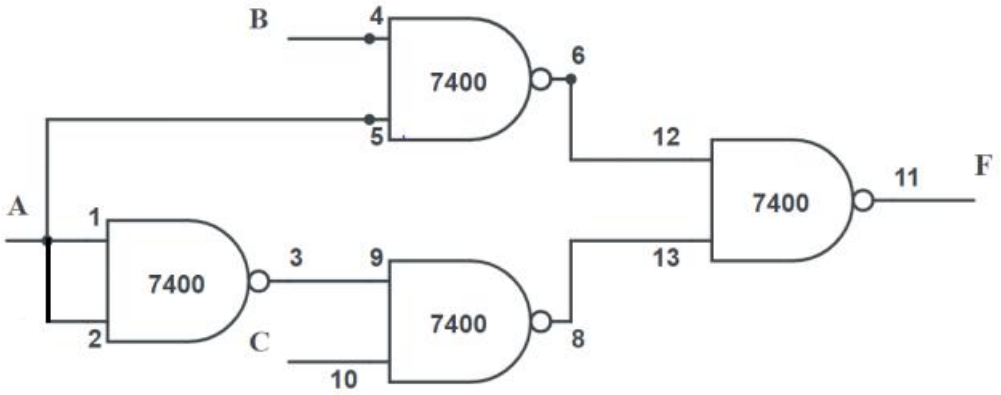

Figure 1.1: SOP implementation of Boolean function using NAND gates.

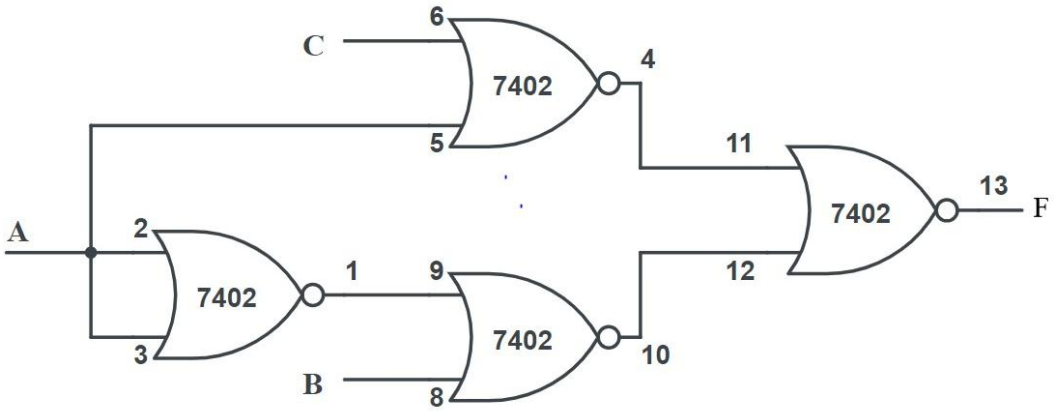

Figure 1.2: POS implementation of Boolean function using NOR gates.

Procedure

For Software Simulation:

a) Create a module with required number of variables and mention it's input/output.
b) Write the description of given Boolean function using operators or by using the built in primitive gates.
c) Synthesize to create RTL Schematic.
d) Create another module referred as test bench to verify the functionality and to obtain the waveforms of input and output.
e) Follow the steps required to simulate the design and compare the obtained output with the corresponding truth table.
f) Take the screenshots of the RTL schematic and simulated waveforms.

Note: Students need to write the Verilog HDL code by their own for which they can refer Appendix - A if required.

For Hardware implementation:

a) Turn off the power of the Trainer Kit before constructing any circuit.
b) Connect **power supply (+ 5 V DC)** pin and **ground** pin to the respective pins of the trainer kit.
c) Place the ICs properly on the bread board in the Trainer Kit.
d) Connect VCC and GND pins of each chip to the power supply and ground bus strips on the bread board.
e) Connect the input and output pins of chips to the input switches and output LEDs respectively in the Trainer Kit.
f) Check the connections before you turn on the power.
g) Apply various combinations of inputs according to truth tables and observe outputs of LEDs.

Observation:

To be written by students

Design Problem:

Design a warning indicator using universal gates when the following conditions apply:

I. Switches A,B,C are on.

II. Switches A and B are on but switch C is off.

III. Switches A and C are on but switch B is off.

IV. Switches B and C are on but switch A is off.

Solution:

First derive the truth table for the given design as shown in the Table 1.3. Then obtain the minimized Boolean expression in SOP and POS form. After getting the Boolean expression draw the logic diagram using NAND and NOR gates.

Inputs			Output
A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$\text{SOP: } F(A,B,C) = m_3 + m_5 + m_6 + m_7$$

$$= \sum (3,5,6,7) = AB + BC + AC$$

$$= \overline{((\overline{AB})(\overline{BC})(\overline{AC}))}$$

$$\text{POS : } F(A,B,C) = M_0\, M_2\, M_4\, M_5 = \prod (0,1,2,4)$$

$$= (A+B)(B+C)(A+C)$$

$$= \overline{((\overline{A+B})(\overline{B+C})(\overline{A+C}))}$$

Table 1.3: Truth table

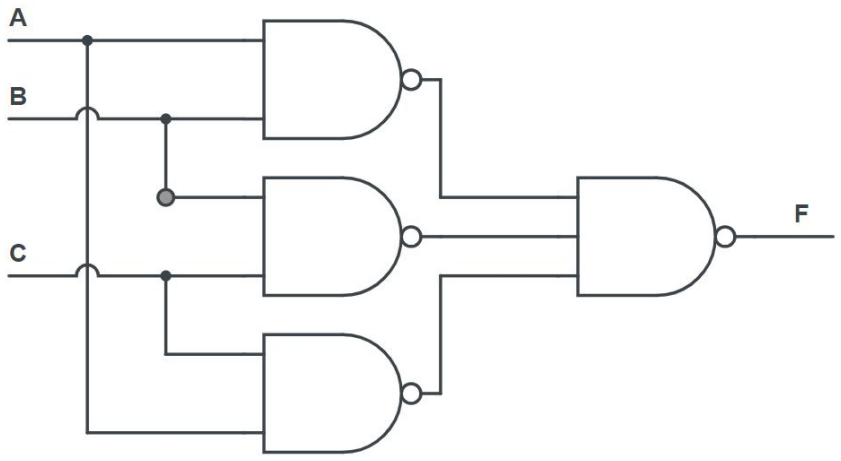

Figure 1.3: Logic diagram of the warning indicator using NAND gates.

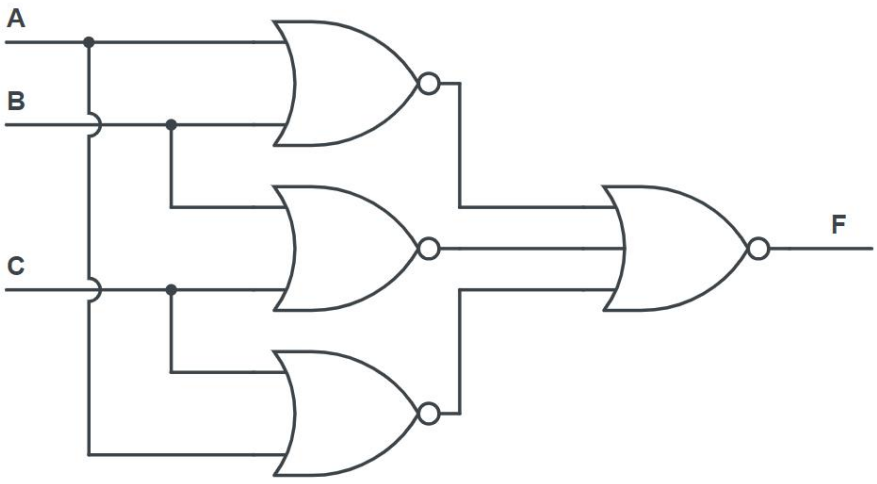

Figure 1.4: Logic diagram of the warning indicator using NOR gates.

Conclusion:

To be written by students.

Sample viva-voice questions

1. What do you mean by logic gates?
2. What are the applications of logic gates?
3. What is DE-Morgan's law?
4. How min-term can be converted to max-term?
5. Design XOR gate using minimum number of NAND gates.
6. Design XNOR gate using minimum number of NOR gates.
7. For the following functions, construct a truth table and draw a circuit diagram using minimum number of universal gates.
   a. F(A,B,C) = AB + A'BC' + AB'C
   b. G(A,B,C) = A'C + ABC + AB'
   c. H(A,B,C,D) = A'BC' + (A $\oplus$ B)C + A'B'CD' + ABC
   d. J(A,B,C,D) = A'C'D' + C'D + CD