EEE 211 / ETE 211 Digital Logic Design
Experiment 1: Digital Logic Gates and Boolean Functions

A. Objectives

- Study the basic logic gates - AND, OR, NOT, NAND, NOR, XOR.
- Acquaint with the representation of Boolean functions using truth tables, logic diagrams and Boolean algebra.
- Prove the extension of inputs of AND and OR gates using the associate law.
- Familiarize with combinational logic circuit.

B. Apparatus

- IC 7400 Quadruple 2-input NAND gates
- IC 7402 Quadruple 2-input NOR gates
- IC 7404 Hex Inverters (NOT gates)
- IC 7408 Quadruple 2-input AND gates
- IC 7432 Quadruple 2-input OR gates
- IC 7486 Quadruple 2-input XOR gates
- Trainer Board

C. Theory

C.1. Digital Logic

Logic Gates

Logic gates are the elementary building blocks of digital circuits. They perform logical operations of one or more logical inputs to produce a single output. Digital logic gates operate at two discrete voltage levels representing the binary values 0 (logical LOW) and 1 (logical HIGH). Table C.1 provides a brief description of the basic digital logic gates, their corresponding IC numbers and circuit symbols.
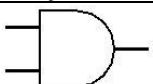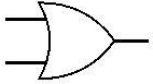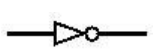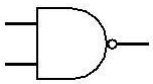
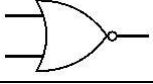| Gate	Description	IC #	Symbol
AND	Multi-input circuit producing an output of 1 if all inputs are 1.	7408	
OR	Multi-input circuit producing an output of 1 when any of its inputs is 1.	7432	
NOT	Single-input circuit that inverts the input (also called an Inverter). The output is 0 if the input is 1 and vice versa.	7404	
NAND	AND followed by an Inverter	7400	
NOR	OR followed by an Inverter	7402	
XOR	The Exclusive-OR or Ex-OR is a two-input circuit that produces an output of 0 is both inputs are same and 1 if the inputs are different.	7486	

Table C.1 Logic gates

Truth Table

A B	$F = A \cdot B$
0 0	0
0 1	0
1 0	0
1 1	1

Table C.2 Truth table of AND

A truth table shows all output logic levels of a logic circuit for every possible combination of inputs. For example, Table C.2 shows the truth table for a two-input AND gate.

Boolean Algebra

Boolean algebra is a branch of mathematical logic that formalizes the relation between variables that take the truth values of *true* and *false*, denoted by 1 and 0 respectively. It is fundamental in the development of digital electronics.Digital electronics networks are generally expressed as Boolean functions. Discrete voltage levels are used to represent the truth values. Postulates and theorems of Boolean algebra are given in Table C.3.

Postulates and Theorems		Name
$A + 0 = A$	$A \cdot 1 = A$	Identity
$A + A' = 1$	$A \cdot A' = 0$	
$A + A = A$	$A \cdot A = A$	
$A + 1 = 1$	$A \cdot 0 = 0$	
$(A')' = A$		Involution
$A + B = B + A$	$AB = BA$	Commutative
$A + (B + C) = (A + B) + C$	$A(BC) = (AB)C$	Associative
$A(B + C) = AB + AC$	$A + BC = (A + B)(A + C)$	Distributive
$(A + B)' = A'B'$	$(AB)' = A' + B'$	De Morgan
$A + AB = A$	$A(A + B) = A$	Absorption

Table C.3 Laws of Boolean algebra

Combinational Logic

Combination logic refers to digital networks where the output is solely dependent on the current input(s) and is not affected by previous states. The analysis of combination logic requires writing the Boolean functions for each element of the circuit, producing their truth tables, and subsequently combining each function for the final output and truth table.

C.2. Integrated Circuits

Integrated Circuits

IC - Integrated Circuit

Figure C.1 illustrates an example IC. The basic rule for most ICs is that there is polarity mark, such as the half-moon notch shown in the figure. Another common polarity mark is a small dot, triangle or tab by pin 1. The rule is to move counter-clockwise around the chip from the polarity mark while numbering the pins starting at 1. Sometimes no direct mark may be present, in which case the pin numbers can be inferred simply from the orientation of the text inscribed on the IC.
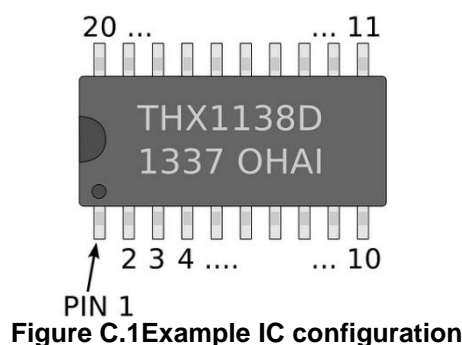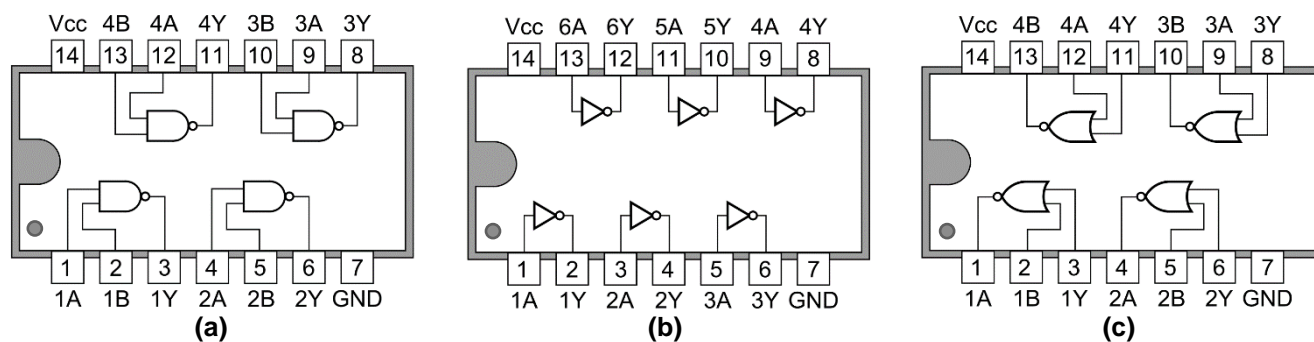

Figure C.1Example IC configuration

7400 Series Integrated Circuits

TTL - Transistor-Transistor Logic

The 7400 series of digital logic ICs represents the most popular family of TTL ICs. Most such modern ICs have been replaced with CMOS. To find the IC number on the chip, simply read the numbers off it ignoring the letters. For example, **74**HC**04**N is the 7404 Hex Inverter IC where the HC denotes it is a high-speed CMOS variant of the TTL circuit.

Figure C.2 shows the pin configurations of the basic logic gate ICs. Figure C.2(a) shows the pin configuration of IC 7400 quadruple 2-input NAND gates. The pin configurations of ICs 7408 AND, 7432 OR and 7486 XOR are same as IC 7400 NAND. Figure C.2(b) and (c) show the pin configurations of ICs 7404 hex inverters and 7402 quadruple 2-input NOR

gates respectively. Note that the input and output pins of the NOR gates are reversed compared to the NAND gates. For all of the above ICs, pin 7 is designated GND (logical LOW) and pin 14 is connected to +5 V as V_{CC} (logical HIGH).

(a)   **(b)**   **(c)**

Figure C.2 Schematic of (a) 7400 NAND, 7408 AND, 7432 OR and 7486 XOR, (b) 7404 NOT, and (c) 7402 NOR ICs

D. Procedure

D.1. Introduction to Basic Logic Gates

1. Place the 7408 AND IC on the breadboard.
2. Connect the V_{CC} and GND pins of the IC to the +5 V and GND ports of the trainer board respectively.
3. Label the pin numbers of the inputs and output of the gate in Figure D.1, using the pin configurations in Figure C.2.

Figure D.1 Pin configurations of gates in ICs

4. Connect the gate.
    a. Connect each input of the gate to a toggle switch on the trainer board.
    b. Connect the output of the gate to an LED on the trainer board.
Apply all combinations of inputs by turning the toggle switches on (1) and off (0), and record if the LED is on (1) or off (0) as the output of the gate. Record your results in
5.   Table D.1.

Input $A\,B$	AND $F = A \cdot B$	OR $F = A + B$	NAND $F = \overline{A \cdot B}$	XOR $F = A \oplus B$	NOR $F = \overline{A + B}$		Input A	NOT $F = \overline{A}$
0 0							0	
0 1							1	
1 0								
1 1								

Table D.1 Truth table of logic gates

6. Replace the AND IC with OR, NAND and XOR ICs without changing the connections and repeat step 5 for each.
7. Repeat steps 1-5 for the NOT and NOR ICs.

D.2. Extension of Inputs of AND and OR

1. Complete the truth table for the 3-input AND gate in Table D.2.

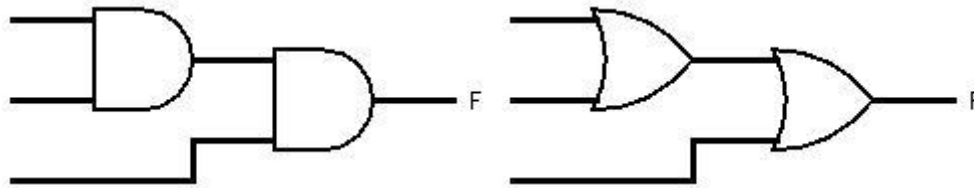| $A\,B\,C$	$F = ABC$	$F = A + B + C$
0 0 0		
0 0 1		
0 1 0		
0 1 1		
1 0 0		
1 0 1		
1 1 0		
1 1 1		

Table D.2 Truth table of 3-input AND and OR

2. Using the associative law given in Table C.3, express the 3-input function using two 2-input AND gates in Table D.3.

$F = ABC =$
$F = A + B + C =$

Table D.3 Associative law

3.   Label the pin numbers in Figure D.2, using the pin configurations in Figure C.2.

Figure D.2 Extension of inputs of AND and OR gates

4.   Connect the circuit.
5.   Connect the output to an LED and verify it using the truth table.
6.   Repeat steps 1-5 for the 3-input OR gate.
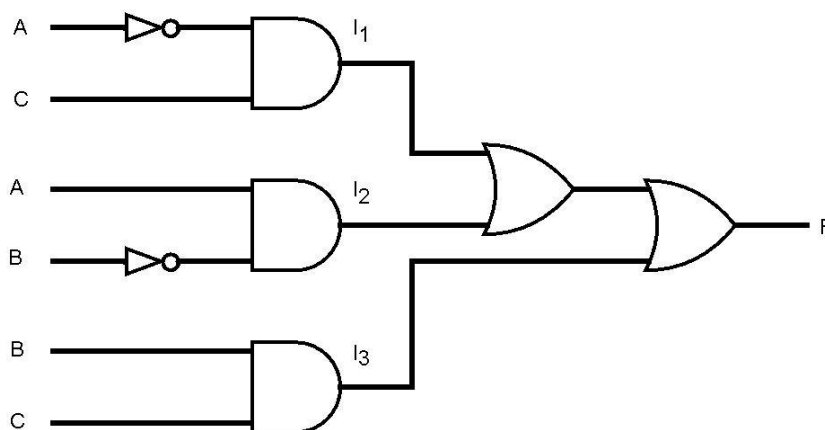
D.3.   Implementation of Boolean Functions

$$F = A'C + AB' + BC$$

1.   Complete the truth table for the *implicants* $I_1 = A'C$, $I_2 = AB'$ and $I_3 = BC$ in Table D.4.
2.   Complete the truth table for the function F in Table D.4.

$A\ B\ C$	$I_1 = A'C$	$I_2 = AB'$	$I_3 = BC$	$F = I_1 + I_2 + I_3$
0 0 0				
0 0 1				
0 1 0				
0 1 1				
1 0 0				
1 0 1				
1 1 0				
1 1 1				

Table D.4 Truth table of logic gates

3.   Label the pin numbers for the NOT, AND and OR gates of the function F in Figure D.3, using the pin configurations in Figure C.2.

Figure D.3 Circuit diagram for the Boolean function F

4.   Connect the input A to a NOT gate using the pins assigned in step 3 and check the output via an LED.
5.   Wire up implicant I_1.
6.   Connect the output of I_1 to an LED and verify it using the truth table.

7. Connect the input B to a NOT gate using the pins assigned in step 3 and check the output via an LED.
8. Wire up implicant I_2.
9. Connect the output of I_2 to an LED and verify it using the truth table.

10. Wire up implicant I_3.
11. Connect the output of I_3 to an LED and verify it using the truth table.

12. Connect the outputs of the three implicants as inputs to the OR gates (using the associative law).
13. Connect the output F to an LED and verify the function using the truth table.

E. Questions

1. Is it possible to make a 3-input NAND or NOR gate with 2-input NAND or NOR gates? Justify your answer.

F. Simulation

1. Simulate the combinational logic circuit of Experiment 0 in Logisim and attach the circuit in your lab report, showing only the instance when the input $ABC = 010$.

ECE.211L 31 May 2015