Hacettepe University Computer Engineering Department

Experiment 1 - Basic Logic Gates

Due date for report and circuit: Week 4 (15/10/2024) - Report until 13:40

BBM233 Digital Design Lab - 2024 Fall

AIM

In this experiment some basic logic gates such as AND, OR, NOT, NAND and NOR gates and some circuits resulting from their combinations will be examined.

BACKGROUND

In digital electronics, Boolean logic refers to the manipulation of binary values in which a 1 represents the concept of *true* and a 0 represents the concept of *false*. In electronic circuits that implement logic, binary values are represented by voltage levels.

In the framework of digital electronic, logic gates give the possibility to perform Boolean Algebra operations on electrical signals. Within this context:

- A logical 1 on the input or output of a gate corresponds to a relatively high voltage value designated as High, Hi or H.
- A logical 0 on the input or output of a gate corresponds to a relatively low voltage value designated as Low, Lo, L.

The basic logic gates are:

- Inverter (NOT gate),
- AND gate, and
- OR gate.

As their names indicate, these gates represent a basic operation in Boolean Algebra. The AND and the OR gates can each have any number of inputs but only one output. An inverter has only one input and one output.

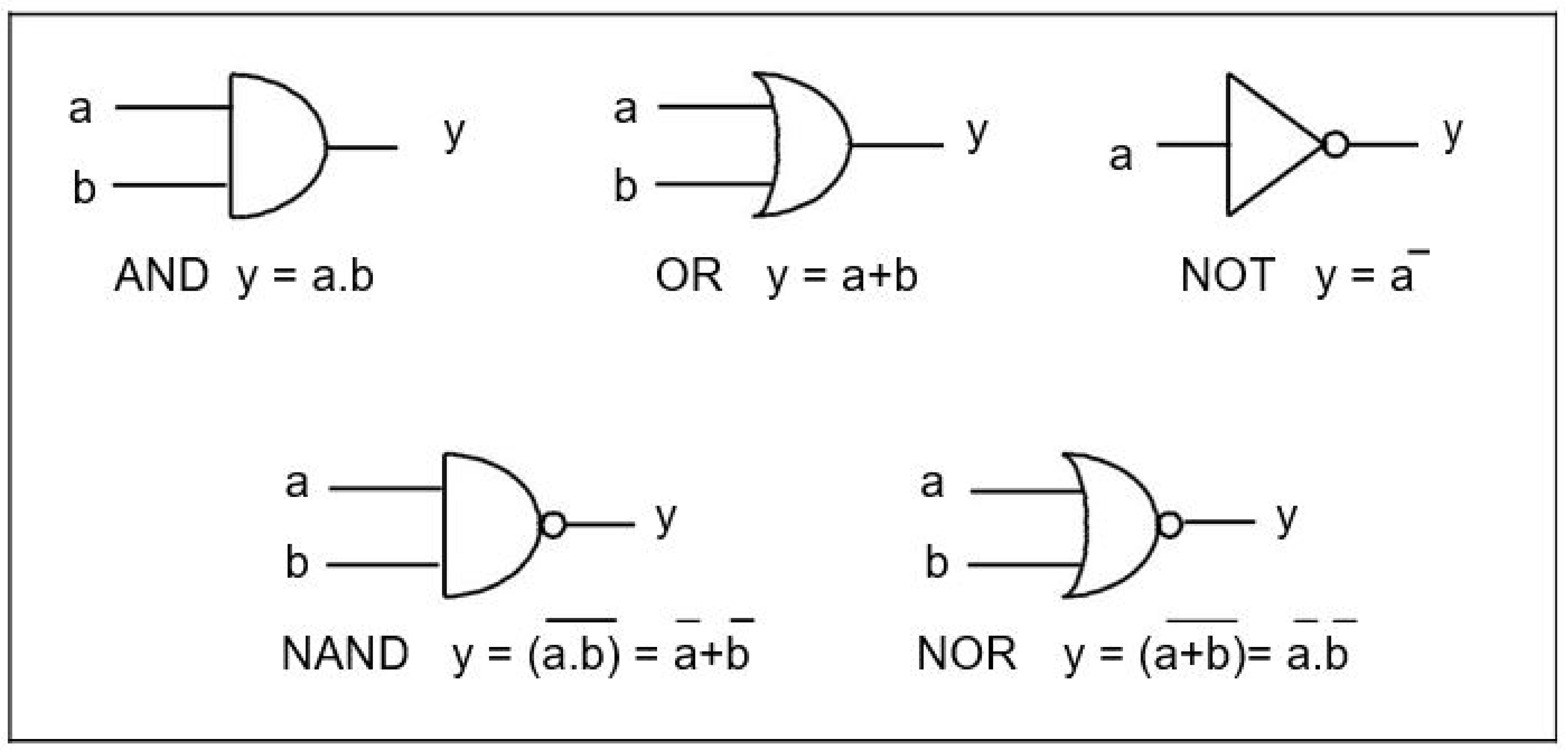


Figure 1: Basic Logic Gates

A NAND gate is equivalent to the combination of an AND gate and an inverter (NOT gate). Therefore the word NAND is a contraction of the words NOT and AND. There is also NOR (NOT-OR) gate which is the concatenation of OR and NOT gates. Because the fabrication of NAND and NOR gates is much easier than the AND and OR's, these two combination gates represent the most popular off-the-shelf packages.

BOARD EXPERIMENT

To ensure the well-being of campus animals, Hacettepe University wants to implement an automated feeding system. This system aims to dispense food efficiently, preventing overfeeding and ensuring animals are fed when necessary. The feeding mechanism must consider scheduled feeding times, the presence of animals, and whether the feeder is empty.

The system has three single-bit inputs:

1. **T (Feeding Time)** - a signal that is **HIGH** during scheduled feeding times and **LOW** at other times.
2. **A (Animals Present)** - a signal from motion sensors. It is **HIGH** when animals are detected near the feeder.
3. **F (Feeder Empty)** - a signal that is **HIGH** when the feeder is empty and **LOW** when it contains food.

T	A	F	Dispense Food
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 1: Truth table for the feeding system

The single-bit output of your digital system controls the feeder mechanism to dispense food. It should be activated (**HIGH**) in the following cases:

- **Scheduled Feeding with Empty Feeder:**
  - When it is scheduled feeding time **and** the feeder is empty, **regardless of animal presence**.
- **Animal-Triggered Feeding Outside Schedule:**
  - When it is not scheduled feeding time, animals are present, **and** the feeder is empty.

Experiment Steps:

1. Fill out the given truth table based on the system specifications provided.
2. Write the Boolean function **Dispense_Food(T, A, F)** in the **sum-of-minterms** form.
3. Implement (draw) the corresponding circuit (e.g., in CircuitVerse) using **NOT**, **AND**, and **OR** gates.

Abbreviations: T = **Feeding Time**, A = **Animals Present**, F = **Feeder Empty**.

Hacettepe University Computer Engineering Department

Experiment 1 - Basic Logic Gates

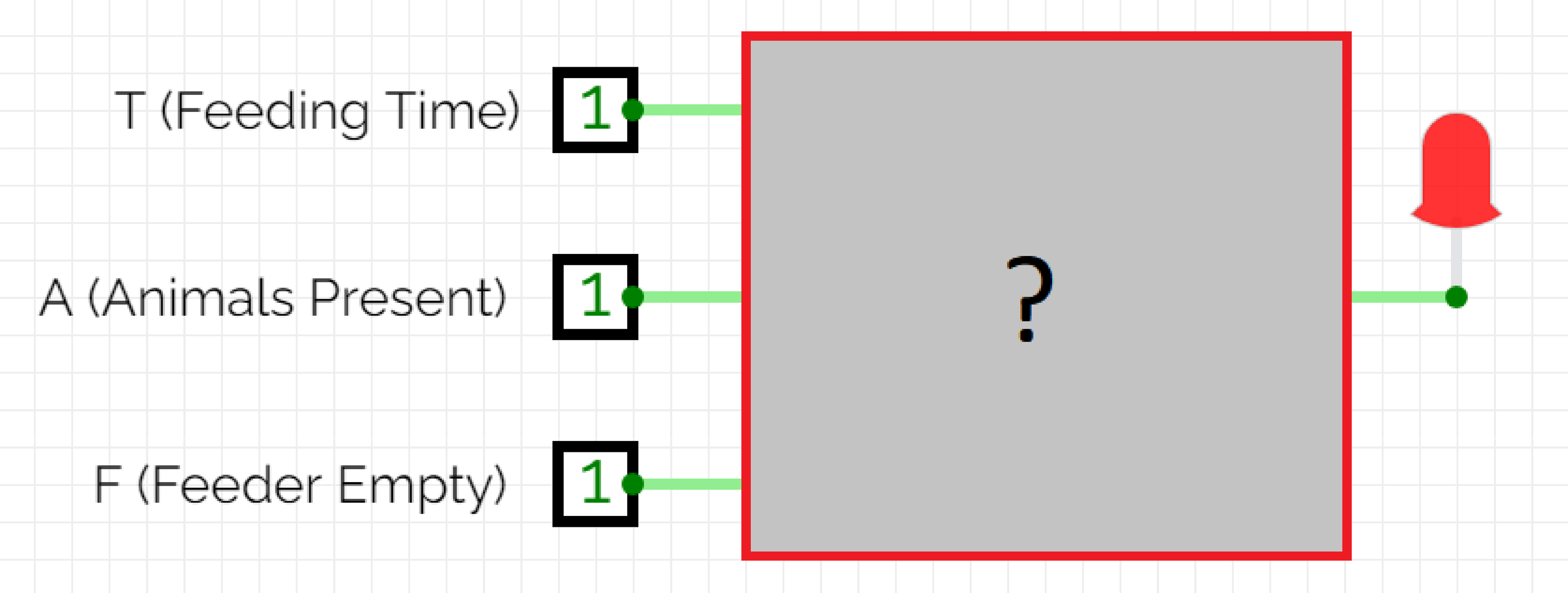
Due date for report and circuit: Week 4 (15/10/2024) - Report until 13:40

BBM233 Digital Design Lab - 2024 Fall

What to Turn In: Report 10%

Your circuit should represent the correct and working implementation of the Boolean function **Dispense_Food(T, A, F)**. Don't forget to test your circuit implementation with all possible input combinations.

Dispense_Food(T, A, F) = <sum-of-minterms Boolean function>



You are encouraged to use this Experiment Report LaTeX Template (click for a PDF preview) and create your reports in LaTeX. We suggest using Overleaf platform for this. This is not mandatory, and you can submit a handwritten report as well, but make sure your report has all the necessary parts and information.

For the full credit, your report must include:

1. The complete truth table,
2. The Boolean function **Dispense_Food(T, A, F)** in the **sum-of-minterms** form (other forms will not get full credit!),
3. Circuit implementation (e.g., circuit drawing by hand or a screenshot of your CircuitVerse implementation).

Submit the report in PDF format through https://submit.cs.hacettepe.edu.tr/
Submission format only zip archives are supported:

- b<StudentID>.zip
  - name_surname.pdf

You must have either a hard copy of your report or be able to show the electronic version at the beginning of the lab. EACH STUDENT MUST SUBMIT THEIR INDIVIDUAL REPORT!

Grading Policy

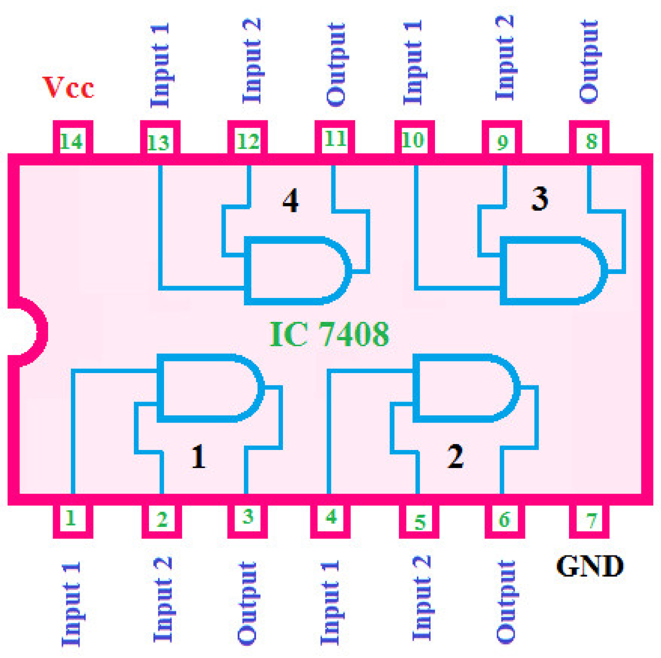
1. **PART 1**: Report (must be shown in the lab) 10%
2. **PART 2**: Correct implementation of the given Boolean function on the breadboard in the lab: 90%

EXPERIMENT PART 2: Implementation 90%

In the lab, you will be given a Boolean function similar to but a bit simpler than the one from your report. You must implement it on the breadboard using **NOT**, **AND**, and **OR** gates on your integrated circuits.

Experiment steps:

1. Make sure that the power supply to the board is turned off.
2. Place the ICs on the breadboard, and connect the GND (0V) and Vcc (5V) to their corresponding pins on each IC (example IC is given in the figure below). Pins 7 must be connected to GND and Pins 14 to Vcc. They must always be connected. Use the IC schematics as a reference to check the pin numbers and what they correspond to.



3. Do not turn on your board during this process as it will give 5V to ICs directly and can burn them.
4. Select switches on the board that will correspond to the function inputs, and an LED to represent the function output.
5. Implement the given Boolean function by connecting the inputs to the correct pins on the ICs (you may choose which gates to use as there are more gates on ICs than you will need). Also connect the outputs of the gates as inputs to other gates as necessary to implement the function. Connect the final output that represents the function output to the LED.
6. When you are certain that everything is connected correctly, power on the board and try your implementation for all input cases to verify its correctness.

Board experiment will be performed in groups of two. Both team members will be graded over the same circuit. Expect to be asked questions about your implementation. You need to answer correctly for the full grade.

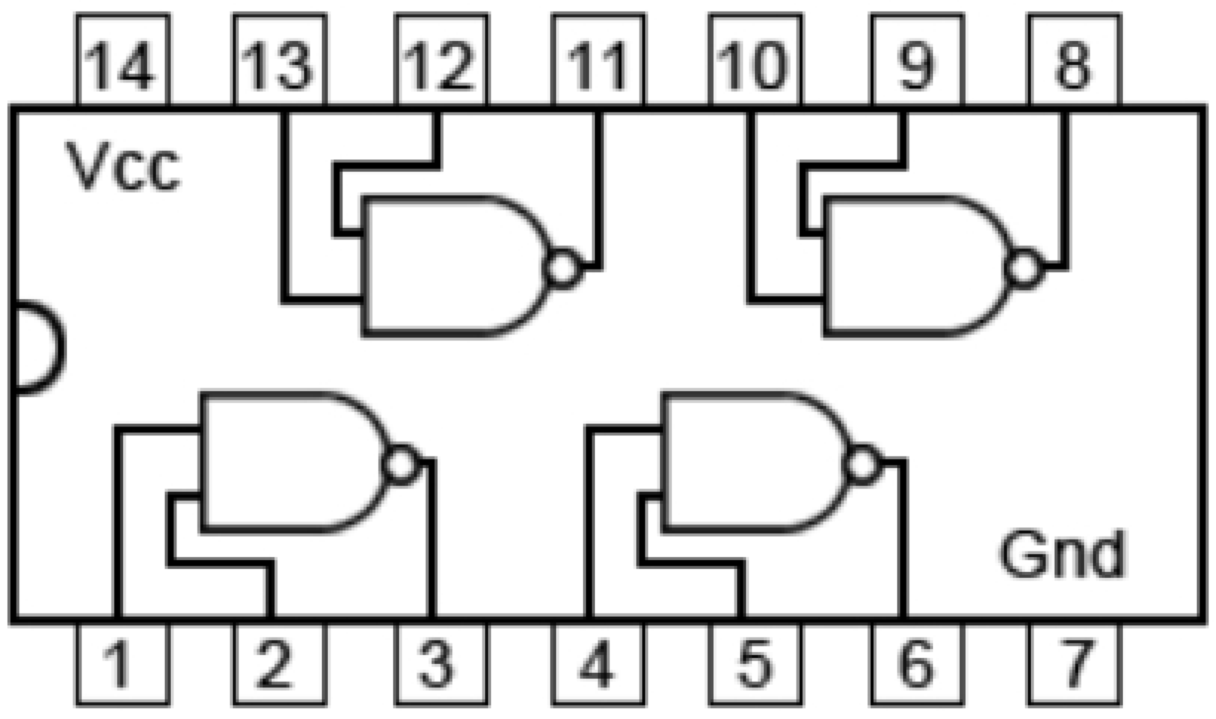
Please, I need your hElp, executTe me

Experiment 1 - Basic Logic Gates

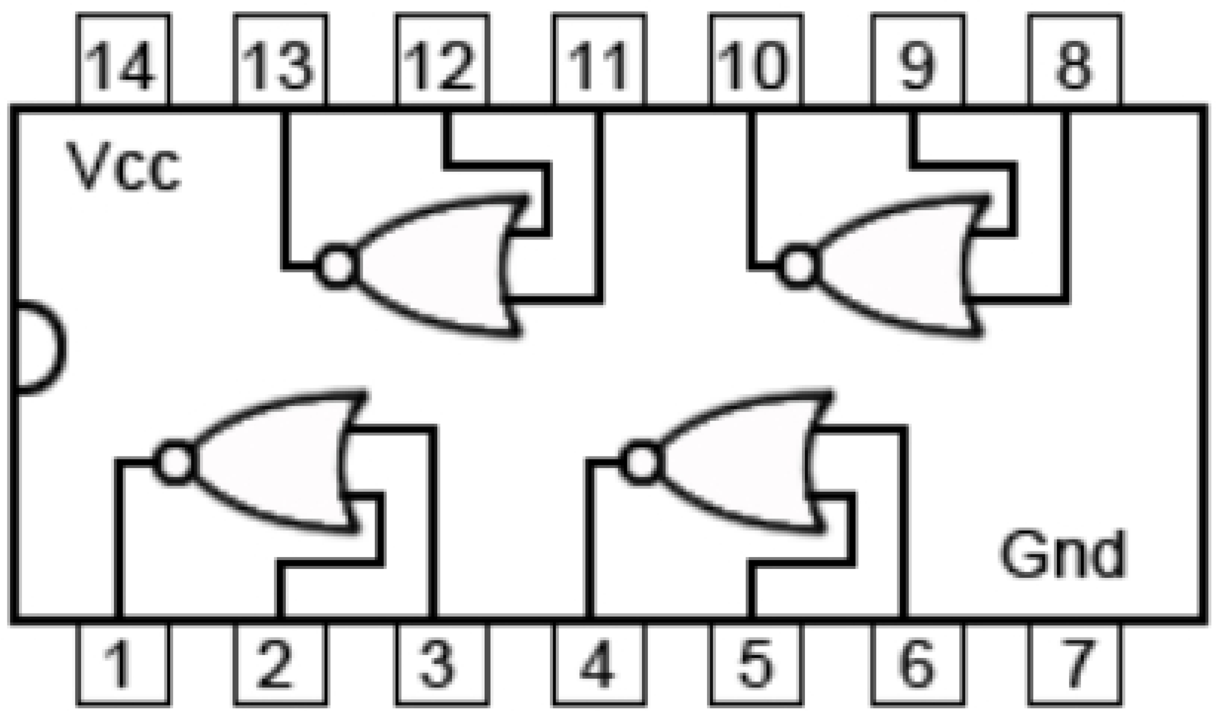
Due date for report and circuit: Week 4 (15/10/2024) - Report until 13:40

BBM233 Digital Design Lab - 2024 Fall

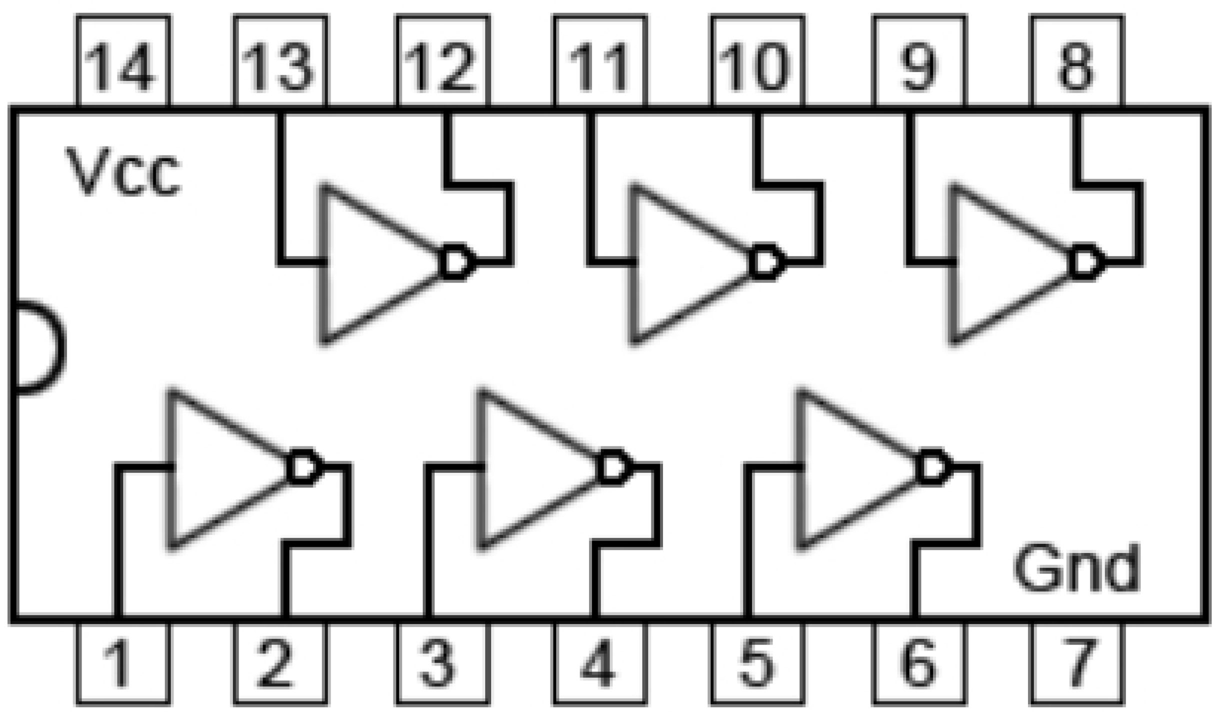
Circuit Diagrams of the Integrated Circuits



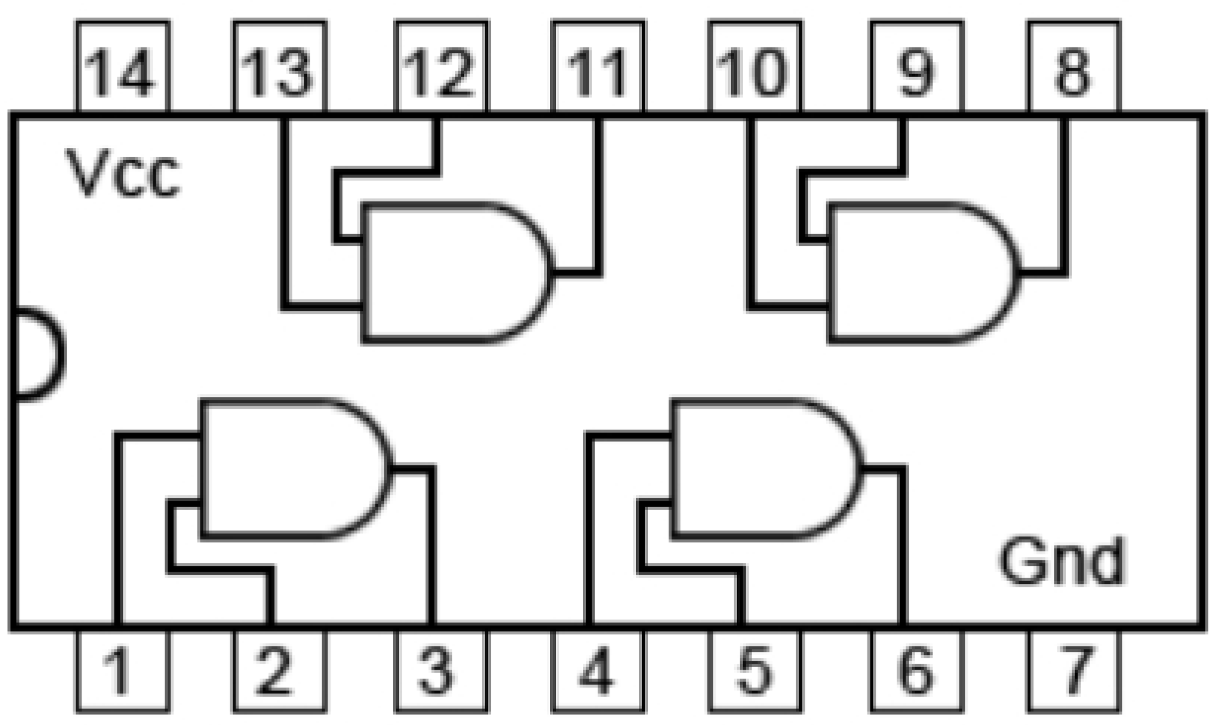
7400 Quad 2 input
NAND Gates



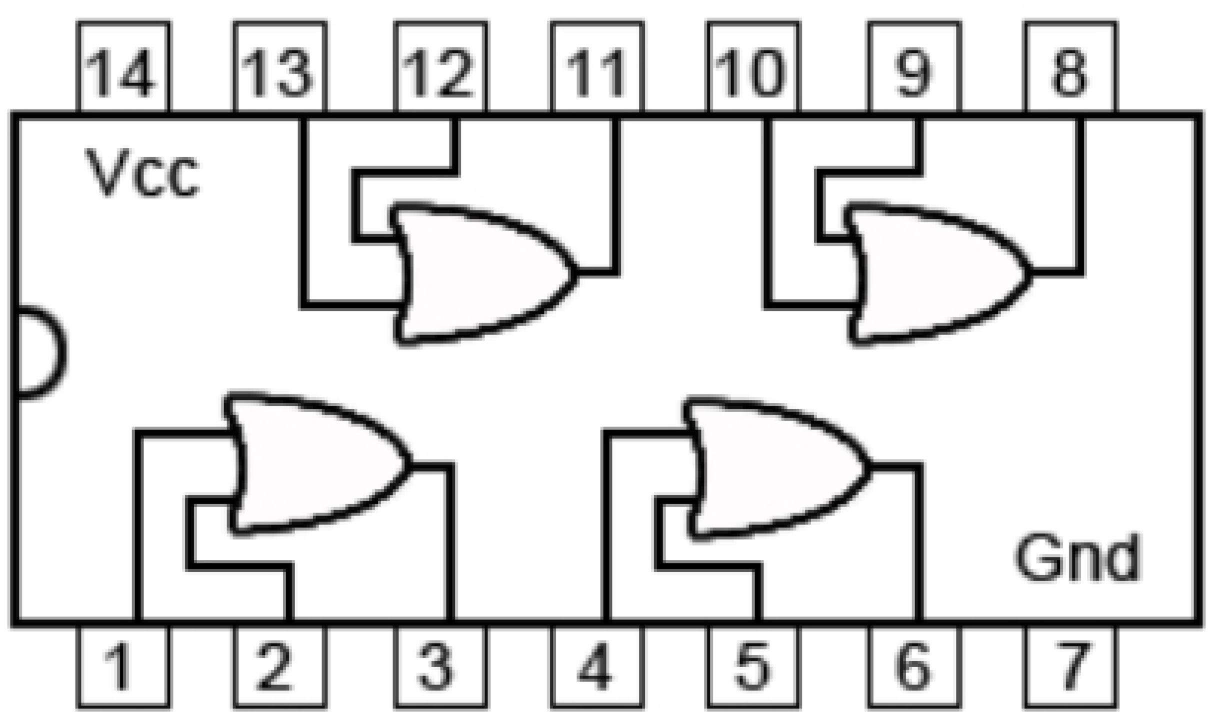
7402 Quad 2 input
NOR Gates



7404 Hex NOT Gates
(Inverters)



7408 Quad 2 input
AND Gates



7432 Quad 2 input
OR Gates

Please, I need your hElp, execuTe me