Foundations of Computer Science

Logic and Gates Lab

Goals:

1. To become familiar with the use of a standard "breadboarding" device
2. To become familiar with Boolean Logic
3. To become familiar with the use of SSI chips.
4. To build and test a half adder

Equipment:

Heath Digital Design Experimenter
Switching diodes
1K and 10K ohm resistors
7408 Quad Dual Input AND chip
7432 Quad Dual Input OR chip
7404 Hex inverter
7486 Quad Dual Input XOR chip

Method:

1. At the beginning of the lab, the instructor will explain the use of the Heath Digital Design Experimenter. The features you should be able to use are: (ask questions if you are unclear)
   - the power supplies
   - the data switches
   - the led's
   - the "breadboard"

The instructor will also explain the use of diodes, resistors and chips.

Answer question 1

2. There are three types of basic Logic Operations. These can be executed by electronic circuits or devices (also called gates or switches). They are: AND, OR and NOT. AND and OR have two inputs and one output. NOT has one input and one output. Using 0's to represent 0 Volts and 1's to represent 5 Volts, the tables below show the outputs for all the possible inputs on the logic gates. Below each table (called a "truth table") is the electronic symbol for the device.

AND Gate

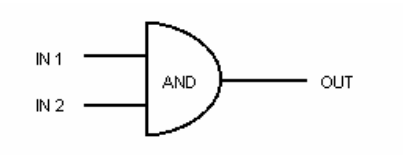
In1	In2	Out
0	0	0
0	1	0
1	0	0
1	1	1

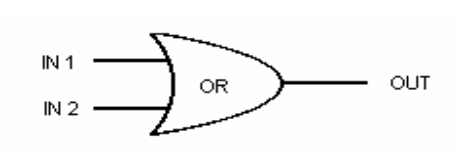
OR Gate

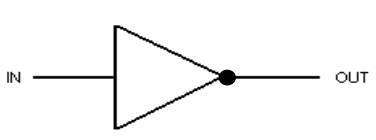
In1	In2	Out
0	0	0
0	1	1
1	0	1
1	1	1

NOT

In	Out
0	1
1	0

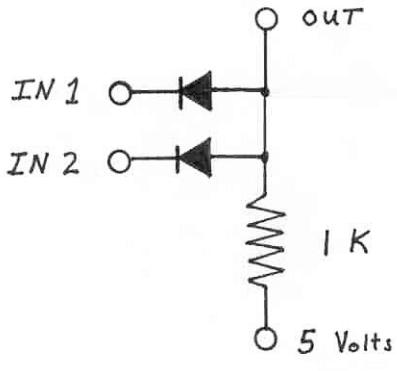




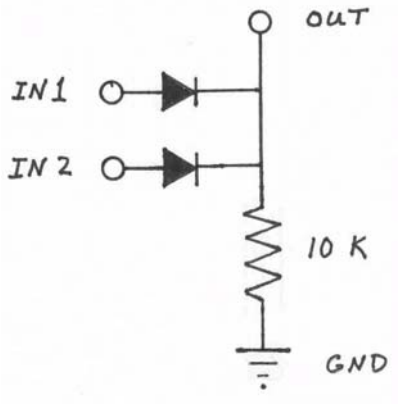


Answer Question 2.

3. The diagrams below are schematic diagrams of an AND gate and an OR gate built from diodes and resistors. Logic devices that use diodes and resistors are called DDL Logic (Diode Diode Logic). Build a DDL AND gate and a DDL OR gate. Test your circuits by experimentally verifying their truth tables.



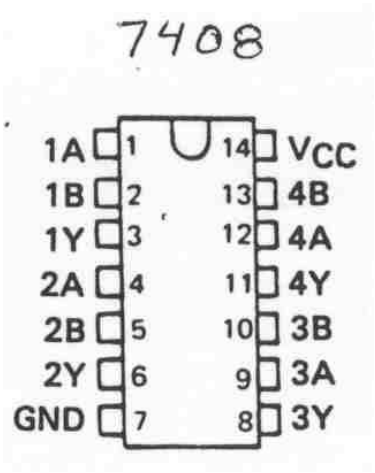
DDL AND Gate

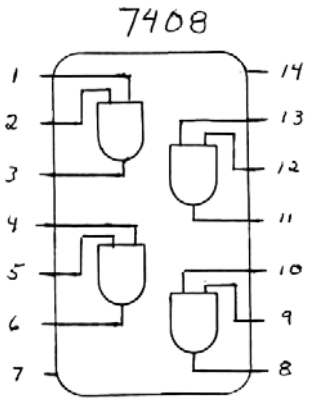


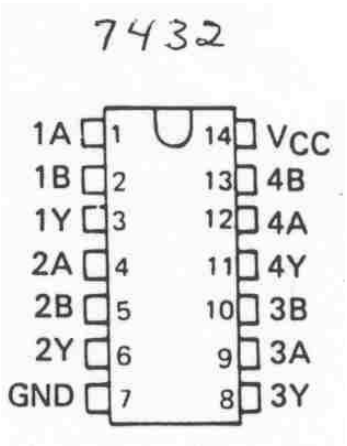
DDL OR Gate

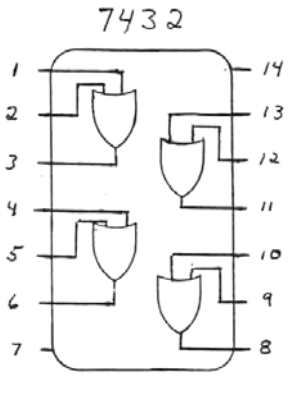
Answer Question 3.

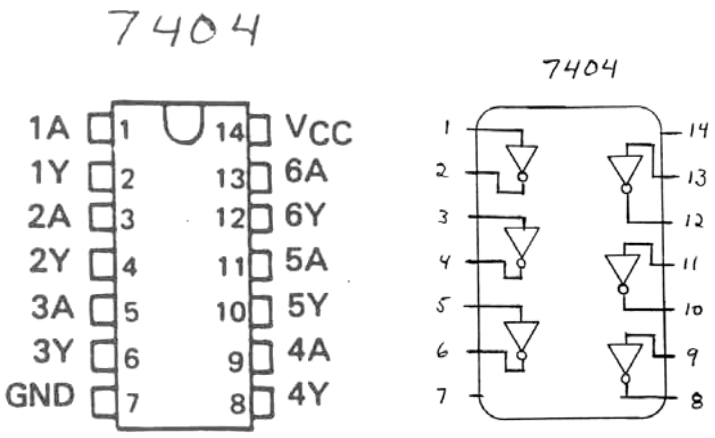
4. An SSI chip (small scale integrated circuit chip) is a device which contains a few gates on a single chip. The various devices can be implemented using many different technologies (TTL, CMOS, ECL, etc.). The standard bipolar TTL series is called the "7400" series (TTL is Transistor-Transistor Logic ... slightly different from DDL but with the same results.). The 7400 series contains about 75 different devices. The 7408 is a quad, dual input AND gate, i.e. it contains 4 AND gates each with two inputs. The 7432 is a quad, dual input OR gate. The 7404 is a hex inverter (i.e. 6 NOT gates on one chip). Below are is the physical and logic diagrams for each chip. The pins marked Vcc must be connected to the +5V power supply and the pin marked ground must be connected to ground to provide power and ground for each of the gates on the chip.







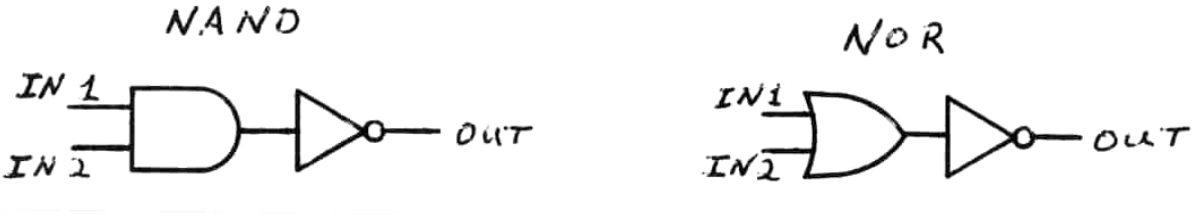




Test one of the AND gates, one of the OR gates and one of the NOT gates by experimentally verifying their truth tables.

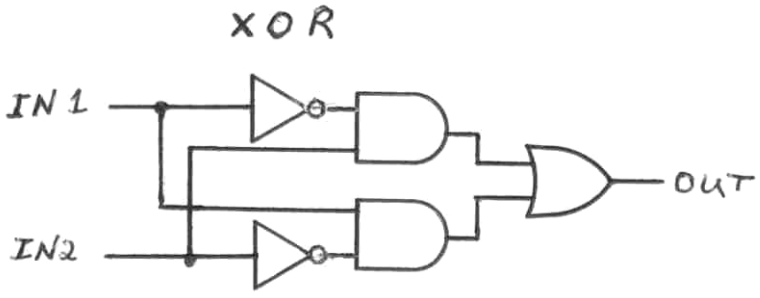
Answer Question 4.

5. If you put the output of an AND gate into the input of a NOT gate the output of the combination is called a NAND gate. Likewise, if you put the output of an OR gate into a NOT, you get a NOR. These are shown below.

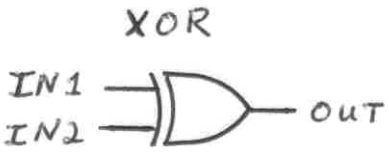


Answer Question 5.

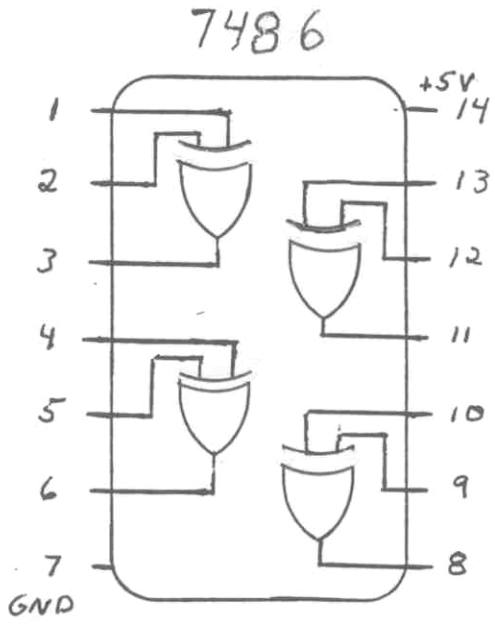
6. Another gate, a very useful gate, is called the Exclusive OR (EOR or XOR). It is made from AND, OR and NOT gates as shown below.



It's logic symbol is:



It is also available as an Quad Dual Input XOR SSL chip, the 7486, as shown below.

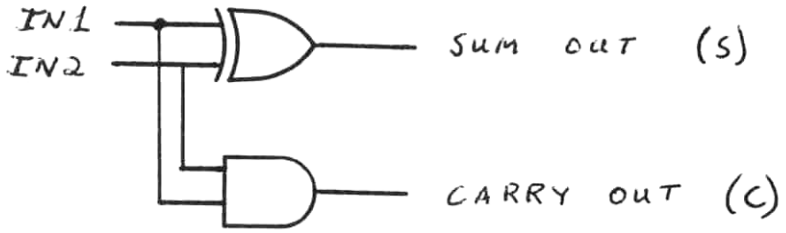


For questions 6 and 7, you may use one of the gates on the 7486 to experiment with the XOR, or you can build the XOR according to the diagram of AND, OR and NOT shown above.

Answer Question 6.

7. Shown below is the logic diagram for a "half adder". It is made by using both an XOR and an AND gate. A half adder is capable of adding any two binary numbers yielding both their "sum" and a "carry" (if necessary). It is called a half adder because it doesn't allow for a carry from a previous addition. An adder that allows a "carry in" is called a "full adder."

   Construct the half adder and verify that it works, i.e. that it correctly adds all possible combinations of binary numbers. (0+0 0+1 1+0 1+1)



Answer Question 7.