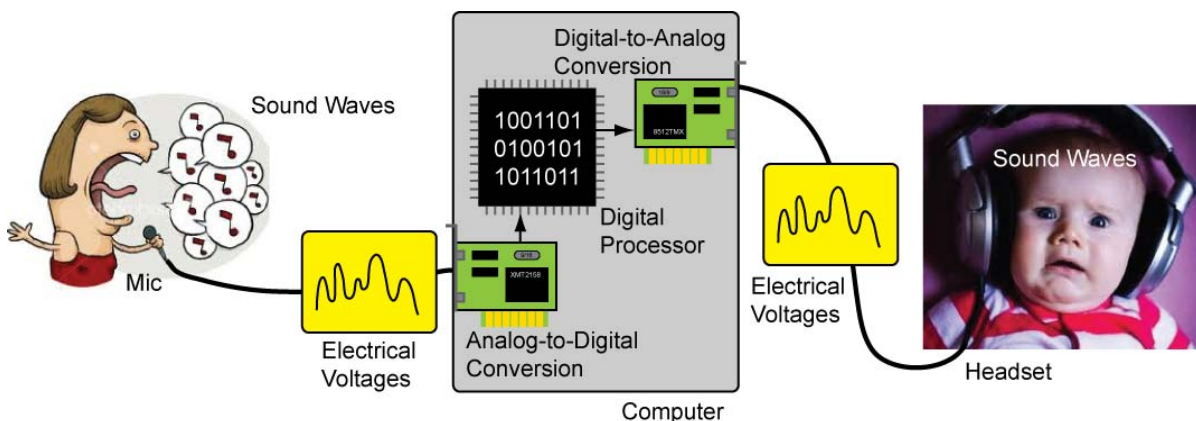


## Electronics and Robotics I Week 12 Logic Gates

- **Administration:**
  - Prayer
- **Electricity and Electronics, Section 20.1, Digital Fundamentals:**
  - **Fundamentals:**
    - Two major fields of electronics:
      - Analog:
        - A system of continuous change without interruption.
        - Example: Potentiometer to control an LED
      - Digital:
        - A system that has discrete values.
        - Example: Digital thermometer or DMM
    - Advantages of digital over analog systems:
      - Less expensive compared to analog systems
      - Information easier to store
      - Digital data can be processed and transmitted more efficiently and reliably
      - Compatible with computer systems
      - Less temperature effect resulting in more stable operation
      - Less sensitive to noise sources.
    - In digital systems, the voltages are limited to the “logic HIGH voltage” ( $V_{dd} = +5V$  for our discussion) and “logic LOW voltage” ( $GND = 0V$ ). Remember that GND (ground) is the universal reference voltage to which all other voltages are measured or compared.
    - In digital systems, the “logic HIGH voltages” ( $V_{dd}$ ) and the “logic LOW voltages” ( $GND$ ) are not just for supplying power to the circuit; they also represent information.
    - Digital signals that are used to move data are a collection of these two voltage states ( $V_{dd}$  and  $GND$ ).
    - Systems that use two state data to move information are commonly known as binary systems.
    - Analog-to-digital converters (ADC)
      - Microphone to ADC to CD (many other steps involved)
    - Digital-to-analog converters (DAC)
      - CD to DAC to speakers



**Figure 18: Sound Converted to Digital Signal Then Converted Back to Sound**  
Images from <http://www.fanpop.com/spots/singing/images/430336/title/sing-photo>  
and <http://www.smothergoose.com/>

- **Decimal Number System:**
  - Based upon us having ten fingers
  - Decimal has 10 numerals (0, 1, 2,3,4,5,6,7,8, and 9).
  - Example: 306

When the symbols for the first digit are exhausted, the next-higher digit (to the left) is incremented, and counting starts over at 0. In decimal, counting proceeds like so:

000,  
 001,  
 002, ...,008  
 009,  
 010, (rightmost digit starts over, and the next digit to the left (in bold) is incremented)  
 011,  
 012, ...  
 ...  
 098,  
 099, (rightmost two digits start over, and next digit to the left (in bold) is incremented)  
 100,  
 101,  
 102, ...

- After a digit reaches 9, an increment resets it to 0 but also causes an increment of the next digit to the left.
- Decimal numbering system is a weighted system.
- That is, the position of each digit in a decimal number is assigned a weight.
- Weight here refers to the relative importance granted to something
- The leftmost digit is the **Most Significant Digit (MSD)** and the rightmost digit is the **Least Significant Digit (LSD)**.
- The MSD in a number is the digit that has the greatest effect on that number. The LSD in a number is the digit that has the least effect on that number.
- The rightmost bit (LSB) in a decimal number has a weight of  $10^0 = 1$ . Each bit to the left increases the weight by a power of ten.

Positive Powers of Ten (Whole Numbers)									
Position of Digit Left to Right	8	7	6	5	4	3	2	1	0
Decimal Weight	$10^8$	$10^7$	$10^6$	$10^5$	$10^4$	$10^3$	$10^2$	$10^1$	$10^0$
Decimal Equivalent	100,000,000	10,000,000	1,000,000	100,000	10,000	1,000	100	10	1

**Table 1: Positive Powers of Ten**

- Recall that:
  - For any integer a,  $a^0 = 1$ .
  - For any integer a,  $a^1 = a$ .
  - $10^4 = 10 \times 10 \times 10 \times 10 = 10,000$ .
- For whole numbers, each position is given a positive power of ten, e.g.  $10^3$ .
- A decimal number is the sum of the weights of each digit.
- Example:

$$\begin{aligned}
 \text{The number } 306 &= (3 \times 10^2) + (0 \times 10^1) + (6 \times 10^0) \\
 &= (3 \times 100) + (0 \times 10) + (6 \times 1) \\
 &= 300 + 0 + 6 \\
 &= 306
 \end{aligned}$$

○ **Binary Numbering System:**

- Introduction: Digital electronic circuits can be in only two states: on or off. This two state system is called binary and is suited for computers. 2-Way switches are simpler than 10-way switches.
- The binary numbering system has only 2 different numerals (0 and 1).
- To distinguish a binary number from a decimal number, the prefix % will be added to a binary number, e.g., %1100111. So 1,100,111 is a decimal number and %1100111 is a binary number (1,100,111 does not equal %1100111).
- Counting in Binary:
  - Binary has only 2 different numerals (0 and 1), unlike decimal which has 10 numerals (0,1,2,3,4,5,6,7,8, and 9).

Counting in binary is similar to counting in any other number system. Beginning with a single digit, counting proceeds through each symbol, in increasing order. Decimal counting uses the symbols **0** through **9**, while binary only uses the symbols **0** and **1**. Each 0 or 1 is a **binary digit**, or bit.

In binary, counting is the same except that only the two symbols 0 and 1 are used. Thus after a digit reaches 1 in binary, an increment resets it to 0 but also causes an increment of the next digit to the left:

%000,  
 %001,  
 %**010**, (rightmost digit starts over, and next digit to the left (in bold) is incremented)  
 %011,  
 %**100**, (rightmost two digits start over, and next digit to the left (in bold) is incremented)  
 %101, ...

From [http://en.wikipedia.org/wiki/Binary\\_numeral\\_system#Counting\\_in\\_binary](http://en.wikipedia.org/wiki/Binary_numeral_system#Counting_in_binary)

- Complete LAB 1 – Counting in Binary
- Binary numbering system is a weighted system,
  - Like a decimal number, the position of each digit in a binary number is assigned a weight.
  - For whole numbers, the leftmost bit is the **Most Significant Bit** (MSB) and the rightmost digit is the **Least Significant Digit** (LSB).
  - For whole numbers, each position is given a positive power of two, e.g.  $2^3$ .
  - The rightmost bit (LSB) in a binary number has a weight of  $2^0 = 1$ . Each bit to the left increases the weight by a power of two.

Positive Powers of Two (Whole Numbers)								
<b>Position of Bit Left to Right</b>	7	6	5	4	3	2	1	0
<b>Binary Weight</b>	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
<b>Decimal Equivalent</b>	128	64	32	16	8	4	2	1

**Table 2; Positive Powers of Two**

(This table only shows eight bit positions. Many more bits may be added to the left if needed.)

- Not all binary codes are weighed. For example, two codes that are unweighed are called the ASCII and 7-segment codes.

- Converting Binary to Decimal:
  - An easy way to convert back and forth from binary to decimal is to use Microsoft Windows Calculator. You can find this program in the Accessories menu of your Programs. To perform the conversion, you must first place the calculator in scientific mode by clicking on the View menu and selecting Scientific mode. Then click on the "Bin" check box and enter the binary number. Now click on the Dec check box and the calculator will convert the binary number to the equivalent decimal number. Decimal to binary conversions may be performed using the reverse process.

- Summary of Binary Conditions:

**On**

**1**

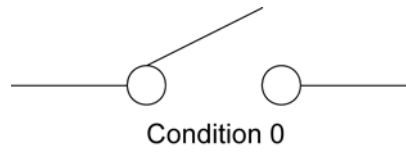
**True**



**Off**

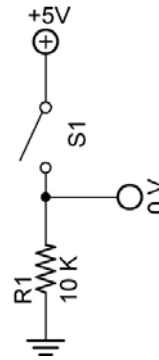
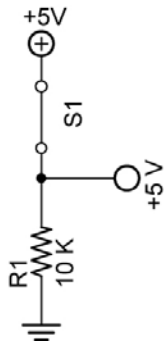
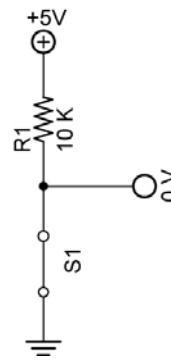
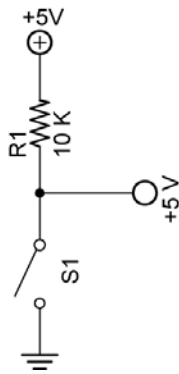
**0**

**False**



**Vdd = +5 Volts (may be less)**

**GND = 0 Volts**

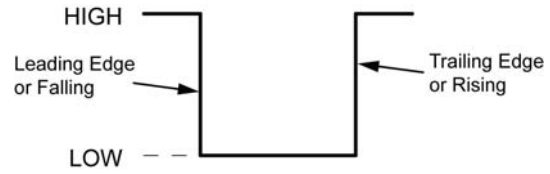
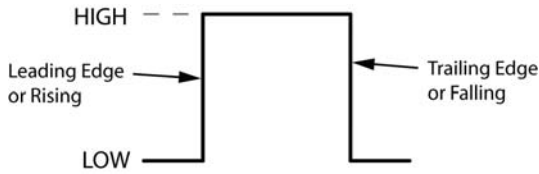


**Figure 1: Binary Conditions Illustrated**

- **Logic Gates:**

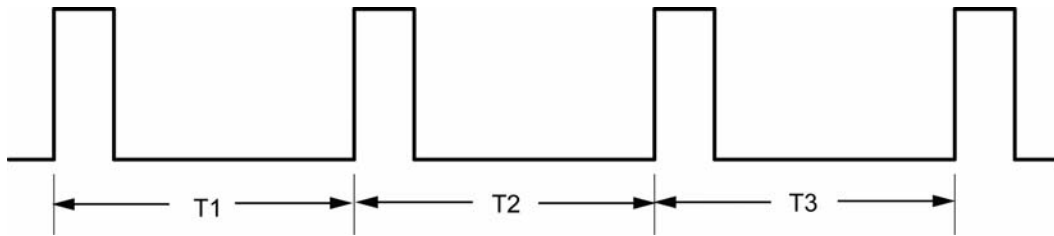
- Digital Waveforms:

- Pulses illustrate the voltage levels (HIGH and LOW) with respect to time.

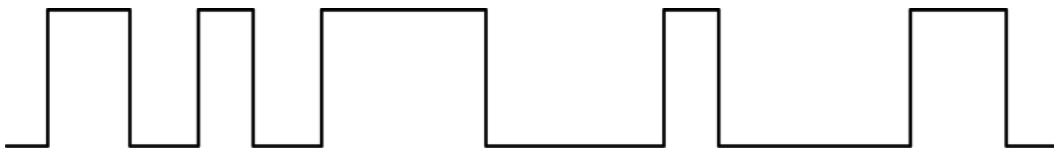


**Figure 2: Positive Going Pulse (Rising)      Figure 3: Negative Going Pulse (Falling)**

- Periodic and Non-periodic Pulses:



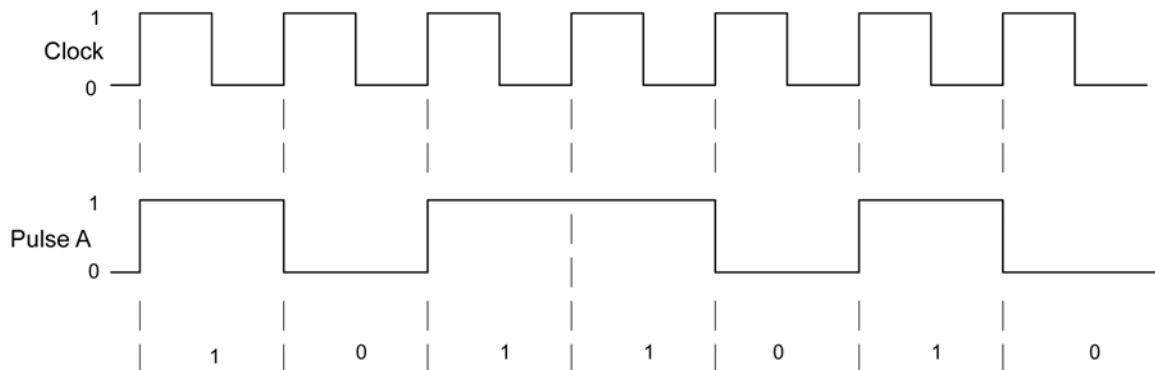
**Figure 4: Periodic Pulse, Period T1 = Period T2 = Period T3**



**Figure 5: Non-Periodic Pulse, Periods Are Not Equal**

- Clock Waveforms:

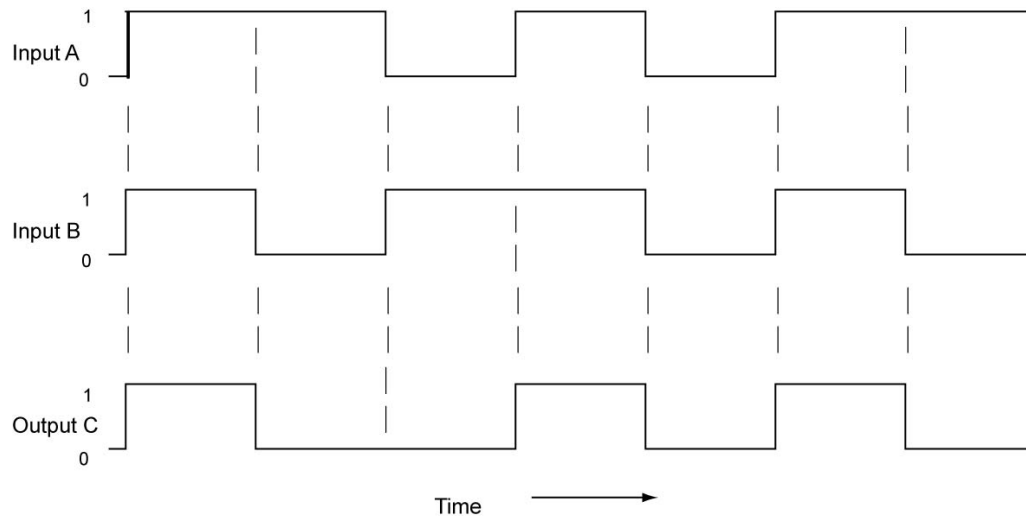
- In digital systems, all waveforms are synchronized by a timing waveform called a clock.
      - A clock waveform is periodic
      - Example:



**Figure 6: Clock Waveform with Associated Waveform A**

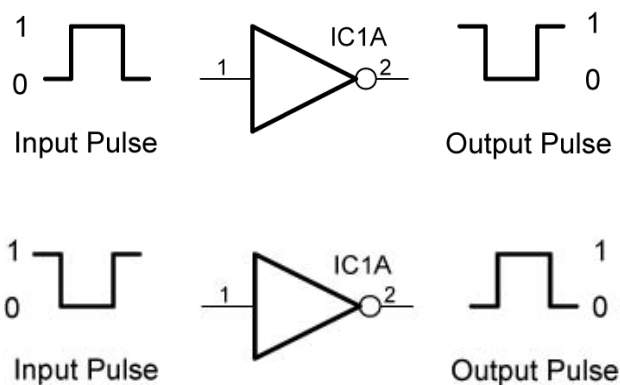
- The clock waveform is a basic timing waveform that synchronizes all waveforms in digital circuits.

- Timing Diagrams: Definition: A timing diagram is a graph of digital waveforms showing the actual time relationship of two or more waveforms and how each waveform changes in relation to each other.
- Example:



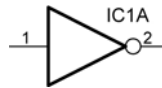
**Figure 7: Timing Diagram for Waveforms A, B, and C**

- Three Fundamental Logic Gates:
  - NOT Gates (Inverters)
  - AND Gates
  - OR Gates
  - These three basic logic gates form the basis for all digital electronic devices, that is, all digital systems; highly complex computer systems can be built entirely of these three basic functions.
- Other Logic Gates:
  - NAND Gates
  - NOR Gates
  - XOR Gates
  - XNOR Gates
- NOT Gates (Inverters):
  - The inverter changes the one logic level to the opposite logic level, that is, it changes a 1 to a 0 and a 0 to a 1.
  - Inputs/Outputs:



**Figure 8: Input and Output Pulses from an Inverter**

- Symbol:



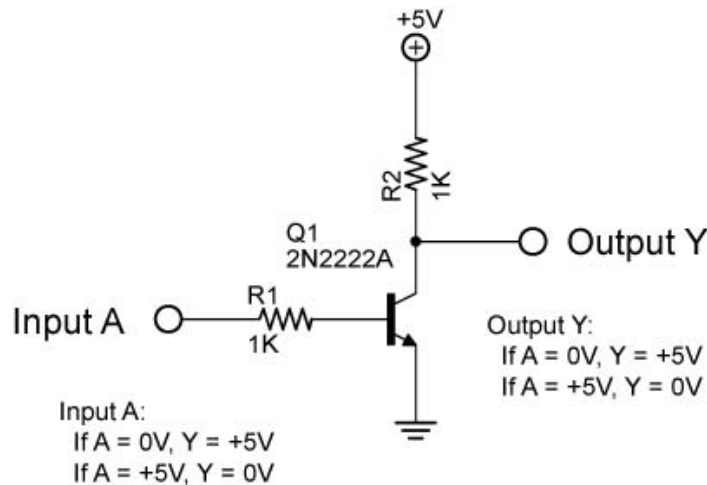
**Figure 9: Inverter (NOT GATE) Symbol**

- Truth Tables:
  - A truth table is the primary tool for displaying the logical relationships between the inputs and outputs in a digital circuit.
  - Every possible combination of inputs is listed in the columns on the left side of the truth table and the corresponding output is listed in the column on the right.
  - A truth table with N inputs has  $2^N$  possible combinations and  $2^N$  rows in the truth table.
  - Truth table for an inverter.

Input	Output
LOW (0)	HIGH (1)
HIGH (1)	LOW (0)

**Table 3: Inverter Truth Table**

- NPN Transistor as an inverter logic gate: The following circuit demonstrates how a NPN transistor may be wired to act as an inverter.

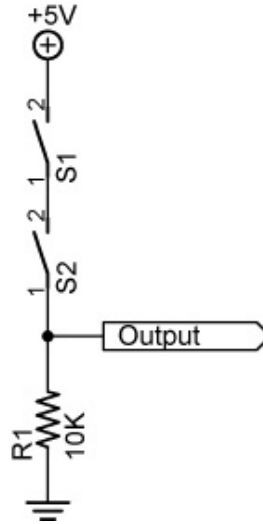


**Figure 10: NPN Transistor Used as an Inverter or NOT Gate**

- When Input A is at 0V, the transistor base-emitter diode is reversed biased turning the transistor off. This is equivalent to an open switch between the emitter and the collector which causes Output Y to +5V.
- When Input A is at +5V, the transistor base-emitter diode is forward biased turning the transistor on. This is equivalent to a closed switch between the emitter and the collector which forces Output Y to same potential as ground, 0V.
- See: <http://elm.eeng.dcu.ie/~digital1/afdez/JavaScript/Page2.htm>
- Perform Logic Gates LAB 2 – NOT Gates (Inverters)
- Complete Question 1 on Class Exercises sheet at the end of the lesson.

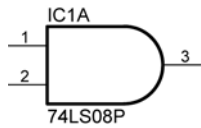


- AND Gates:
  - The AND gate outputs a HIGH only when all of the inputs are HIGH.
  - “All or nothing” feature: The output will be HIGH only if all of the inputs are HIGH, otherwise the output will be nothing or LOW (0V).
  - The purpose of an AND gate is to determine when certain conditions are simultaneously true.
  - “AND Gate” Using Switches:
    - Switches can be configured in series to duplicate the function of an AND gate. In the circuit below, the output is HIGH only when both switches are closed (a 1 condition). If either or both of the switches is opened (a 0 condition), then the output goes LOW.

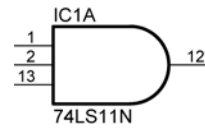


**Figure 11: A 2-Input AND Gate Using Two Switches**

- AND Gate Symbol:



**Figure 12: 2-Input AND Gate**



**Figure 13: 3-Input AND Gate**

- AND Gate Truth Tables:

Inputs		Output
A	B	
0	0	0
0	1	0
1	0	0
1	1	1

HIGH=1, LOW=0

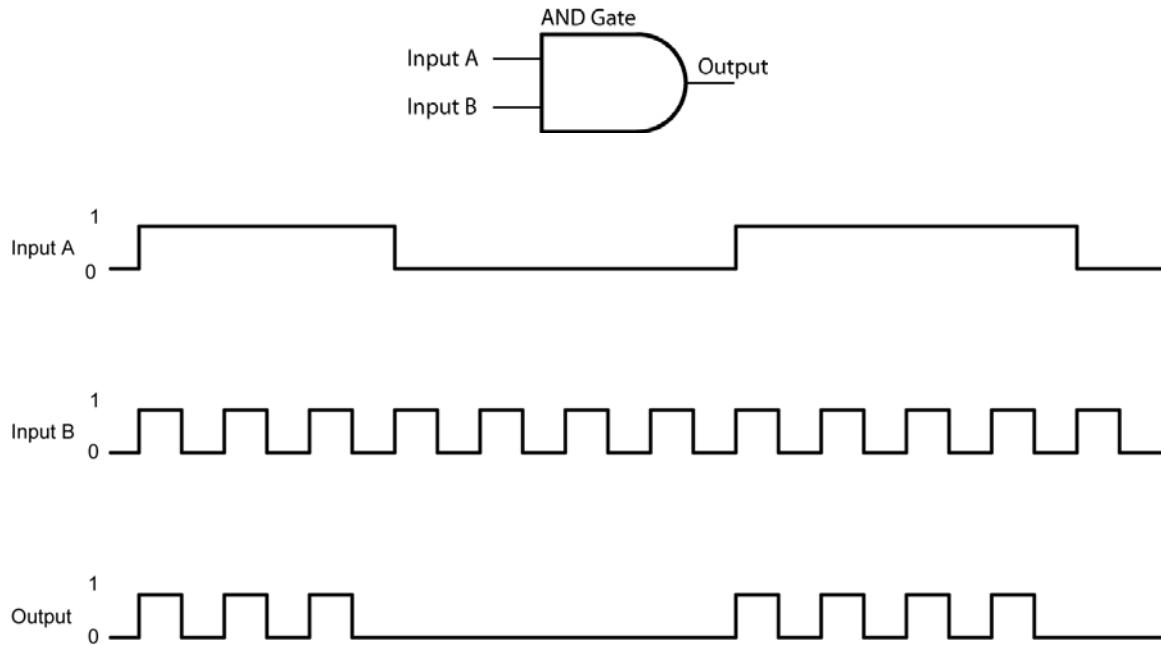
**Table 4: 2-Input (A, B) AND Gate Truth Table**

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

**Table 5: 3-Input (A,B,C) AND Gate Truth Table**

- See: <http://tams-www.informatik.uni-hamburg.de/applets/hades/webdemos/10-gates/00-gates/and.html>

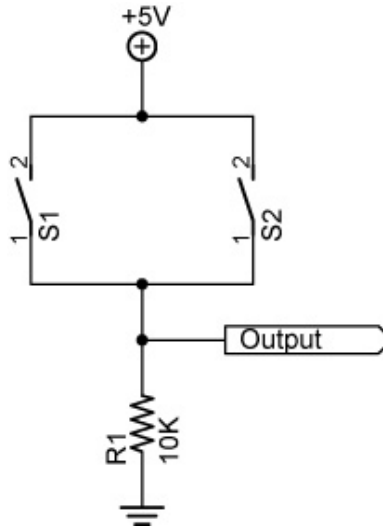
- AND gate as a data valve:
  - One of the AND gate inputs, A, may be used as a valve to permit data into the other input, B, to pass through to the output. See below.



**Figure 14: Timing Diagram Illustrating AND Gate as a Valve**

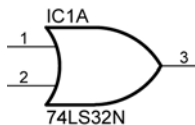
- Data into input B is only allowed to pass to the output when input A is HIGH.
  - Complete Question 2 on Class Exercises sheet.
  - Complete LAB 3 – AND Gates
  - Complete LAB 4 – AND Gates and NOT Gates

- OR Gate:
  - “Anything” feature
  - An OR gate outputs a HIGH when any of the inputs are HIGH.
  - “OR Gate” Using Switches:
    - Switches can be configured in parallel to duplicate the function of an OR gate. In the circuit below, the output is HIGH if either or both of the switches is closed (a 1 condition). If both of the switches are opened (a 0 condition), then the output goes LOW.



**Figure 15: 2-Input OR Gate Using Two Switches**

- OR Gate Symbol:



**Figure 16: 2-Input OR Gate**

- OR Gate Truth Table:

Inputs		Output
A	B	
0	0	0
0	1	1
1	0	1
1	1	1

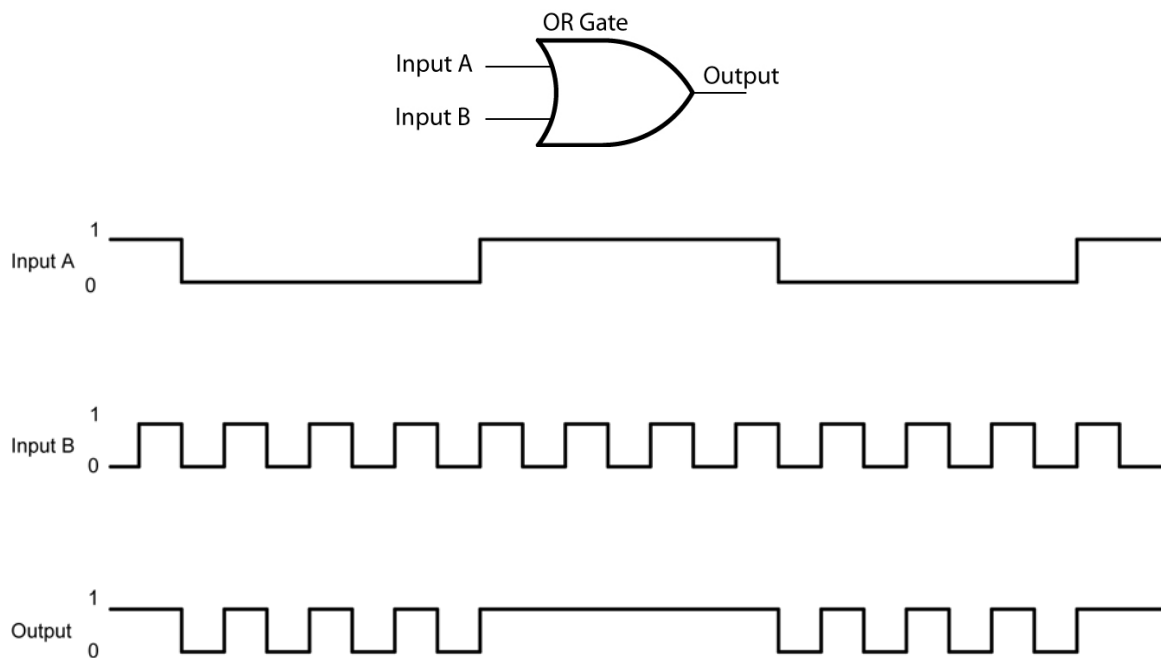
HIGH=1, LOW=0

**Table 6: 2-Input (A, B) OR Gate Truth Table**

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

**Table 7: 3-Input (A,B,C) OR Gate Truth Table**

- An OR gate may be used as a data valve similar to the AND gate. See below.



**Figure 17: Timing Diagram Illustrating OR Gate as a Valve**

- Data into input B is only allowed to pass to the output when input A is LOW
  - See: <http://tams-www.informatik.uni-hamburg.de/applets/hades/webdemos/10-gates/00-gates/or.html>
  - Complete Question 3 on Class Exercises sheet.
- **Continue Building the Robotic Sonar Car:**
  - **Make sure** when mounting the motors to the motor mounts that the mounting screws **are not** too long so as to impact the gears in the gearhead motor.

## Electronics and Robotics I Week 12 Logic Gates LAB 1 – Counting in Binary

- **Purpose:** The purpose of this lab is to develop the student’s skill in counting in binary.
- **Materials:**
  - 1 - Pencil
- **Procedure:**
  - In Table 1, fill in the binary equivalent for the decimal given:
  - In Table 2, fill in the next binary number if you are counting:
- **Results:**

Decimal Number	Binary Number
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	

**Table 1**

Binary Number	%1001
Next Binary Number	
Binary Number	%0001101
Next Binary Number	
Binary Number	%100100111
Next Binary Number	
Binary Number	%00001111
Next Binary Number	
Binary Number	%10001001010
Next Binary Number	
Binary Number	%1111111111
Next Binary Number	

**Table 2**

## Electronics and Robotics I Week 12 Logic Gates LAB 2 – NOT Gates (Inverters)

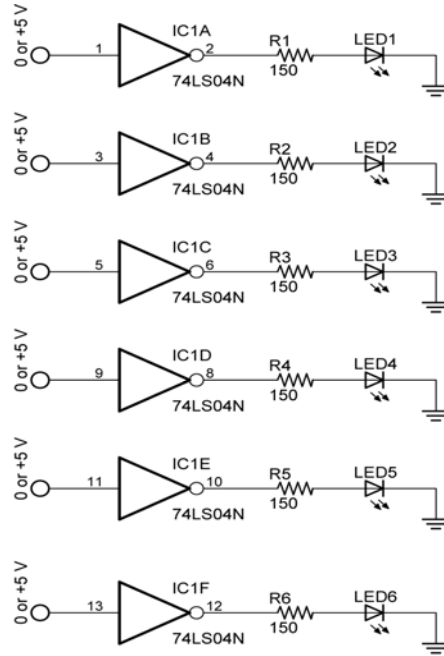
**Purpose:** The purpose of this lab is to acquaint the student with a Hex-Inverter

**Materials:**

- 1 – Analog/Digital Trainer or Breadboard
- 1 – 74LS04N Hex-Inverter
- 1 – 150 Ohm DIP Resistor Package (For Breadboard Only)
- 6 – LEDs (For Breadboard Only)

**Procedure:**

Wire the circuit below. See the photos on the next page.  
Connect the six inputs to the HI/LOW toggles on the analog/digital trainer.  
The LEDs on the analog/digital trainer may be used for LED1-LED6;  
R1-R6 may be eliminated in this case.



Pin 7 to Ground

Pin 14 to +5 V

**Results:**

Input Pin	State	Output Pin	State
1	HIGH	2	
3	LOW	4	
5	LOW	6	
9	HIGH	8	
11	HIGH	10	
13	LOW	12	

Photo of layout on the analog/digital trainer using discrete LEDs:

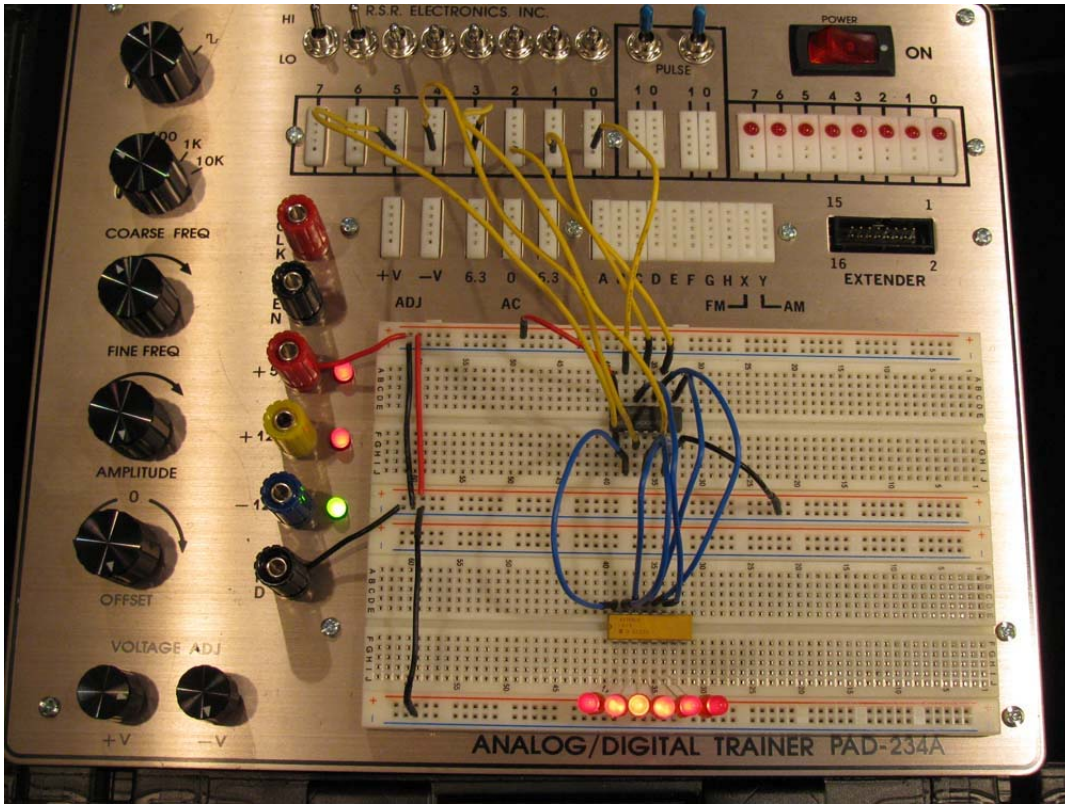
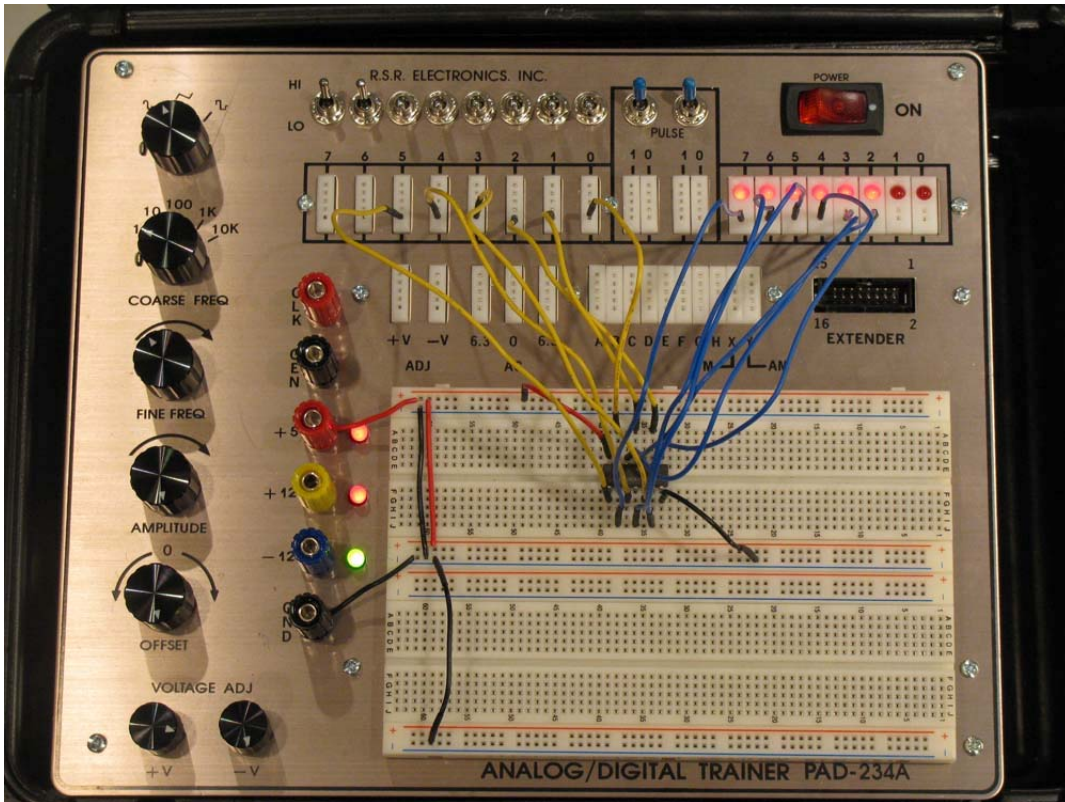


Photo of layout on the analog/digital trainer using LEDs on trainer:





## Electronics and Robotics I Week 12

### Logic Gates LAB 3 – AND Gates

**Purpose:** The purpose of this lab is to challenge the student to become acquainted with the basic operation of an AND gate.

**Materials:**

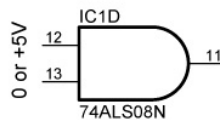
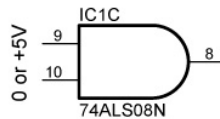
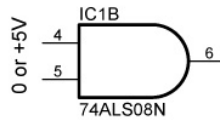
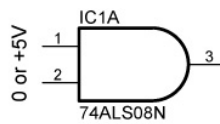
- 1 – Analog/Digital Trainer
- 1 – 74LS08, 2 – Input AND Gate

**Procedure:**

Connect the eight inputs to the HI/LOW toggles on the analog/digital trainer.

Use the LEDs on the analog/digital trainer as the outputs.

Fill in the table in the result section.



Pin 7 to GND, Pin 14 to +5V

### Quad 2-Input AND Gate

**Results:**

Pin	HIGH/LOW
1	LOW
2	LOW
3	
4	LOW
5	HIGH
6	
9	HIGH
10	LOW
8	
12	HIGH
13	HIGH
11	

## Electronics and Robotics I Week 12

### Logic Gates LAB 4 – AND Gates and NOT Gate

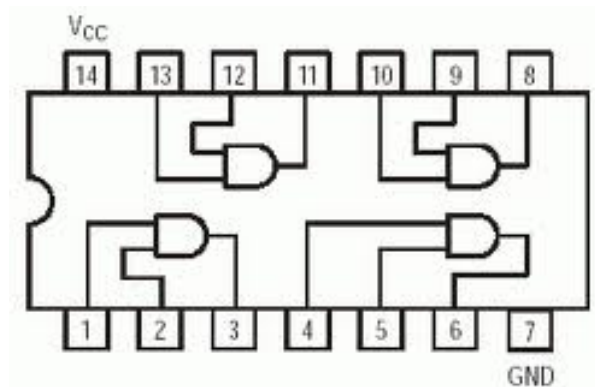
**Purpose:** The purpose of this lab is to challenge the student to solve a real life design problem using hex-inverter (NOT gate) and an AND gate.

#### Materials:

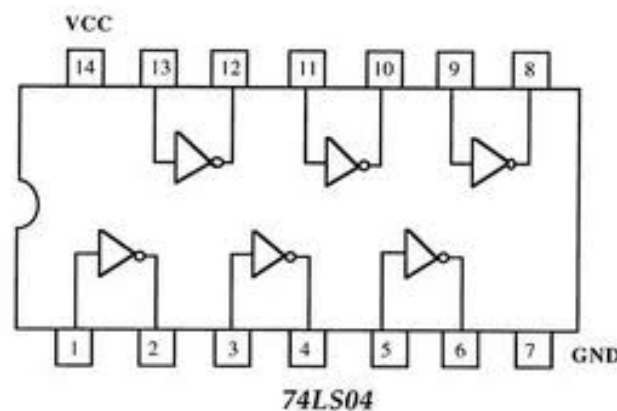
- 1 – Analog/Digital Trainer
- 2 – SPDT Switches (for ignition and seat belt switches)
- 1 – 74LS08, Quad 2-Input AND Gate
- 1 – 74LS04 Hex-Inverter (NOT Gate)
- 1 – Piezo Buzzer

#### Procedure:

- Design and build a circuit using a hex-inverter and an AND gate to simulate a seat belt alarm. The alarm (piezo buzzer) must turn on only when the following conditions are met:
  - When the ignition switch is on **and**
  - When the seat belt switch is off (the seat belt is unbuckled)
- Use the HI/LOW toggle switches for the two switches.



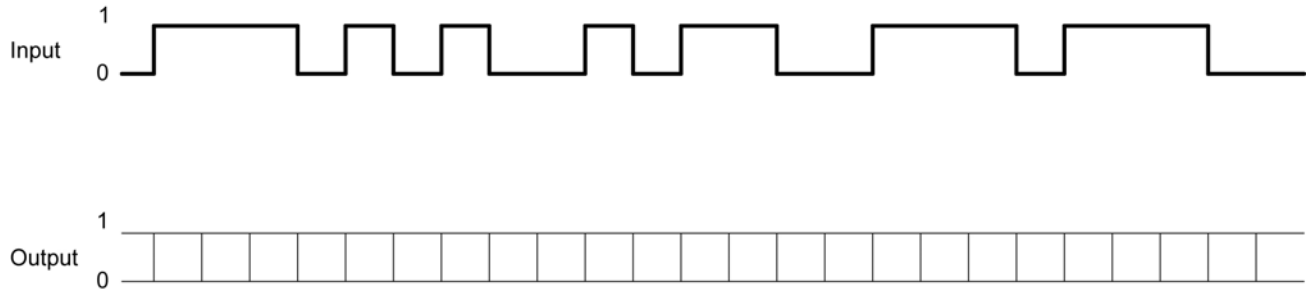
**74LS08 Quad 2-Input AND Gate Pinout**



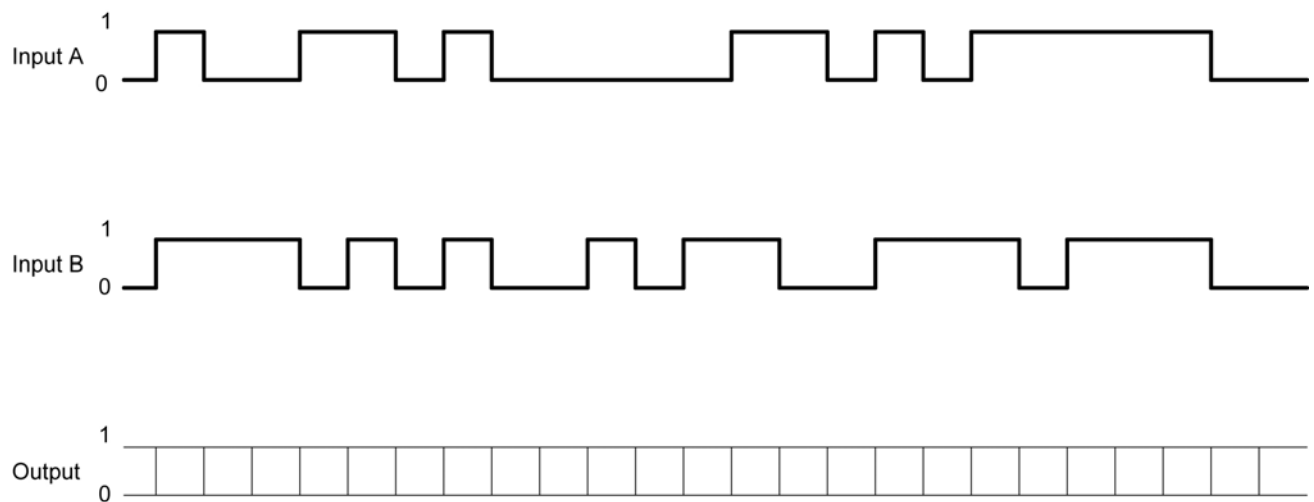
**74LS04 Hex Inverter Pinout**

## Class Exercises

- Draw the output waveform for a NOT gate with the input shown below:



- Draw the output waveform for an AND gate having the inputs shown below:



- Draw the output waveform for an OR gate having the inputs shown below:

