Laboratory 10

Digital Circuits - Logic and Latching (modified from lab text by Alciatore)

Required Components:

- 1x 330Ω resistor
- 4x 1kΩ resistor
- 2x 0.1μF capacitor
- 1x 2N3904 small signal transistor
- 1x LED
- 1x 7408 AND gate IC
- 1x 7474 positive edge triggered flip-flop IC
- 1x 7475 data latch (bi-stable latch) IC
- 3x SPDT switches or NO buttons

Objectives

In this laboratory exercise you will use TTL (transistor-to-transistor logic) integrated circuits (ICs) to perform combinational and sequential logic functions. Specifically, you will learn how to use logic gates and flip-flops. You will use these components to build a simple circuit to control the display of an LED based on the past and current state of various switches or buttons. You will also learn how to read manufacturer TTL data books that summarize the functionality and specifications for a whole family of digital devices.

Introduction

The ICs you will be handling in this laboratory exercise require digital inputs and produce digital outputs. A binary digital signal is a sequence of discrete states, in contrast to an analog signal that varies continuously. Figure 10.1 shows the difference between digital and analog signals. The sampled digital data is a discrete representation of the analog signal. The data is represented by a series of bits.

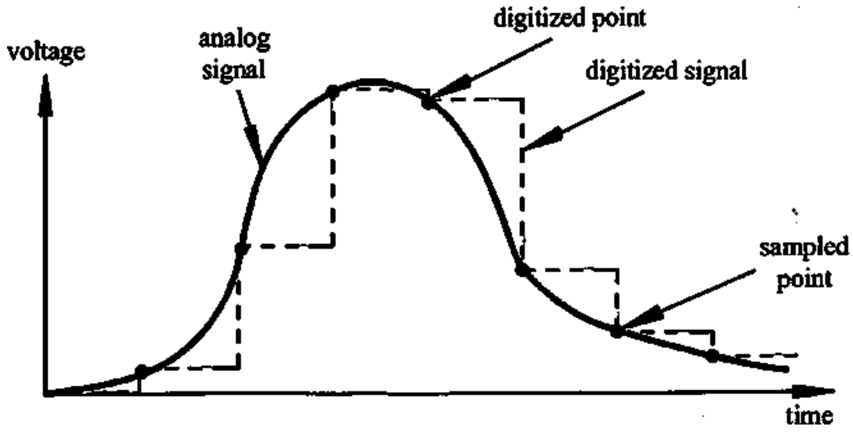


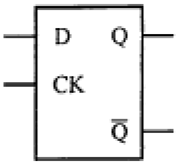
Figure 10.1 Analog and Digital Signals

A binary (digital) signal may exist in only one of two states defined as a voltage high and low. Many types of devices are available for processing the information contained within a digital signal (i.e., a sequence of 0s and 1s). The ICs you will see in this laboratory exercise are TTL (Transistor-Transistor Logic) circuits. TTL devices process digital signals that have a high level defined from 3.5 to 5 V and a low level between 0 to 0.7 V. Note that 0.7 V to 3.5 V is a dead zone. Usually, but not always, a voltage high is equivalent to a logic high. Each of the signals at the input and output terminals of a digital device can exist in only one of two possible states, a voltage low corresponding to a binary zero, or a voltage high corresponding to a binary one.

There are hundreds of TTL ICs (also called chips) available, each with its own functionality. There are many companies that manufacture ICs, but they all use a standard numbering method to identify the ICs. While each chip manufacturer publishes a set of data books that describe these are largely obsolete and datasheets are easily available online. Datasheets all contain the same basic information: chip pin-outs, truth tables, operating ranges, and chip-specific details. Chips in the TTL series use standard naming with a number starting with “74” searching online for the gate you need (e.g., the 2-input AND gate used in this lab) will yield a chip number, e.g., 74LS08 or 74AC08 which are made by Fairchild and have 4 gates (Quad) on each chip. For most purposes, the only numbers that are important are the "74," which corresponds to the standard TTL series, and the "08," which identifies the chip function (in this case, a Quad 2-Input AND). A standard "Quad 2-Input AND Gate" can be referred to simply as a "7408" for any manufacturer.

Data Flip-flops and Latches

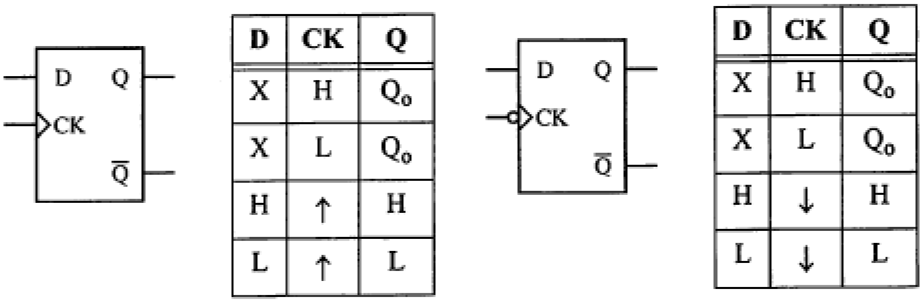
There are many digital circuit applications where you may need to store data for later use. One way to do this is through the use of flip-flops. The bistable data latch (see Fig. 10.2) is a flip-flop that is useful in many applications. The data latch has a data input (D), a clock input (CK), and output Q. Most flip-flops include complementary outputs where Q is the inverse of Q. With a data latch, the data input gets passed to or blocked from the output depending upon the clock signal. When CK is high, Q=D (i.e., the output tracks the input). When CK is low, the D input is ignored and the last value of Q (the value of D when CK last went low) is stored (latched). This memory state of the flip-flop (when CK is low) is indicated in the truth table with Q_0 (the last value latched). The entire functionality is summarized in the truth table shown in Fig. 10.2. An X in a truth table indicates that a signal value may have either value (H or L). For example, for the data latch, when CK is low, the input D has no effect on the output Q.



D	CK	Q
H	H	H
L	H	L
X	L	Q_0

Figure 10.2 Data Latch (7475).

The data latch is sometimes referred to as a level-triggered device since it is active (or triggered) based on the level (high or low) of the clock input, in this case high. A more common type of triggering for flip-flops is edge triggering where the output can change state only during a transition of the clock signal. Devices that respond when the clock transitions from low-to-high (indicated by an up arrow in a truth table) are referred to as positive edge triggered devices. Devices that respond when the clock transitions from high-to-low (indicated by a down arrow in a truth table) are referred to as negative edge triggered devices. Fig. 10.3 summarizes the functionality of positive and negative edge-triggered D-type flips-flops. Positive edge triggering is indicated by a triangle at the clock input. Negative edge triggering is indicated by an inversion circle and triangle at the clock input.



D	CK	Q
X	H	Q_0
X	L	Q_0
H	↑	H
L	↑	L

D	CK	Q
X	H	Q_0
X	L	Q_0
H	↓	H
L	↓	L

Figure 10.3 Positive (left) and Negative (right) Edge-triggered D flip-flop. The 7474 is Positive Edge-triggered.

Figures 10.4 through 10.6 show pin-out and schematic diagrams from the datasheets for various TTL devices used in this exercise. Note that the 7408 includes four AND gates numbered 1 through 4 (e.g., 1Y = 1A • 1B). The 7474 includes two positive edge-triggered data flip-flops, and the 7475 includes four positive level-triggered data flip-flops (AKA "data latches"). **Note that the 7474 has preset and clear features. Because these features are active low, these pin should be connected to 5V to deactivate them.**

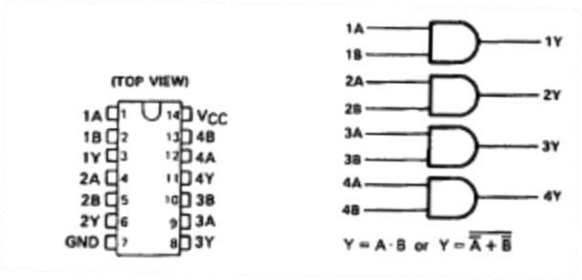


Figure 10.4 Pin-out and schematic symbol diagrams for the 7408

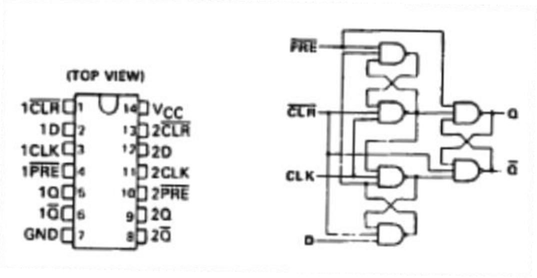


Figure 10.5 Pin-out and schematic symbol diagrams for the 7474

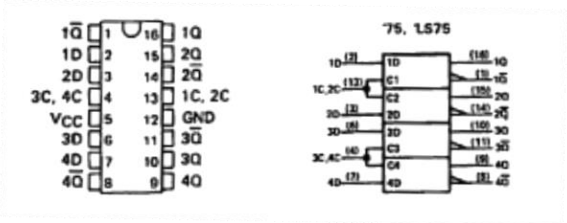


Figure 10.6 Pin-out and schematic symbol diagrams for the 7475

Hints on assembling and troubleshooting breadboard circuits containing integrated circuits

Please follow the protocol listed below when using breadboards to construct and test prototype circuits containing integrated circuits (ICs). Generally, if you carefully follow this protocol you will avoid a lot of frustration

- Start with a clearly drawn schematic illustrating all components, inputs, outputs, and connections.
- Draw a detailed wiring diagram, using the information from handbooks regarding device pin-outs. Label and number each pin used on each IC and fully specify each component. This will be your wiring guide.
- Double check the functions you want to perform with each device.
- Insert the ICs into your breadboard, and select appropriately colored wire (i.e. red for +5V, black for ground, other colors for signals).
- Wire up all connections, overwriting the wiring diagram with a red pen or highlighter as you insert each wire. Use appropriate lengths (~ 1/4") for exposed wire ends. If the ends are too short, you might not establish good connections; and if too long, you might damage the breadboard. Also be careful to not insert component (e.g., resistor and capacitor) leads too far into the breadboard holes. This can also result in breadboard damage.
- Double check the +5V and ground connections to each IC.
- Set the power supply to +5V and turn it off.
- Connect the power supply to your breadboard and then turn it on.
- Measure signals at inputs and outputs to verify proper functionality.
- If your circuit is not functioning properly, go back through the above steps in reverse order checking everything carefully. If you are still having difficulty, use the beep continuity-check feature on the multimeter to verify all connections.
- When removing ICs from the breadboard, use a chip-puller tool to limit the potential for pin damage.
- See more useful information and guidance in Appendix

Laboratory 10 Procedure / Summary Sheet

Names:_________________________________

1. Go online to look up the specifications for the 7408 Quad-AND, the 7474 positive edge-triggered flip-flop, and the 7475 bistable data latch. Pay particular attention to the pin-out diagrams. (**Note - these are also provided at the beginning of this Lab**).
2. Using the datasheet pin-out diagrams, draw a complete, detailed wiring diagram (showing all connections and all pin numbers) for the circuit shown in Fig. 10.8 using a 7474 positive edge-triggered flip-flop. Carefully **label and number all pins that are used on each IC, including power and ground**. Then construct the circuit, being sure to **connect 5V and ground to both ICs** (otherwise, they won't function). You will need to submit your detailed wiring diagram with your Lab summary and answered questions at the end of the Lab (see Question 4). **Note that the 7474 has preset and clear features. Because these features are active low, these pin should be connected to 5V to deactivate them**.

   Note: it is good practice to include a 0.1μF capacitor across the power and ground pins of each IC (not shown in Figs. 10.7 and 10.8). This helps filter out transients that could occur on the power and ground lines during switching. The capacitors are especially important in more complicated circuits where a single power supply may be providing reference voltages and switched current to numerous components.

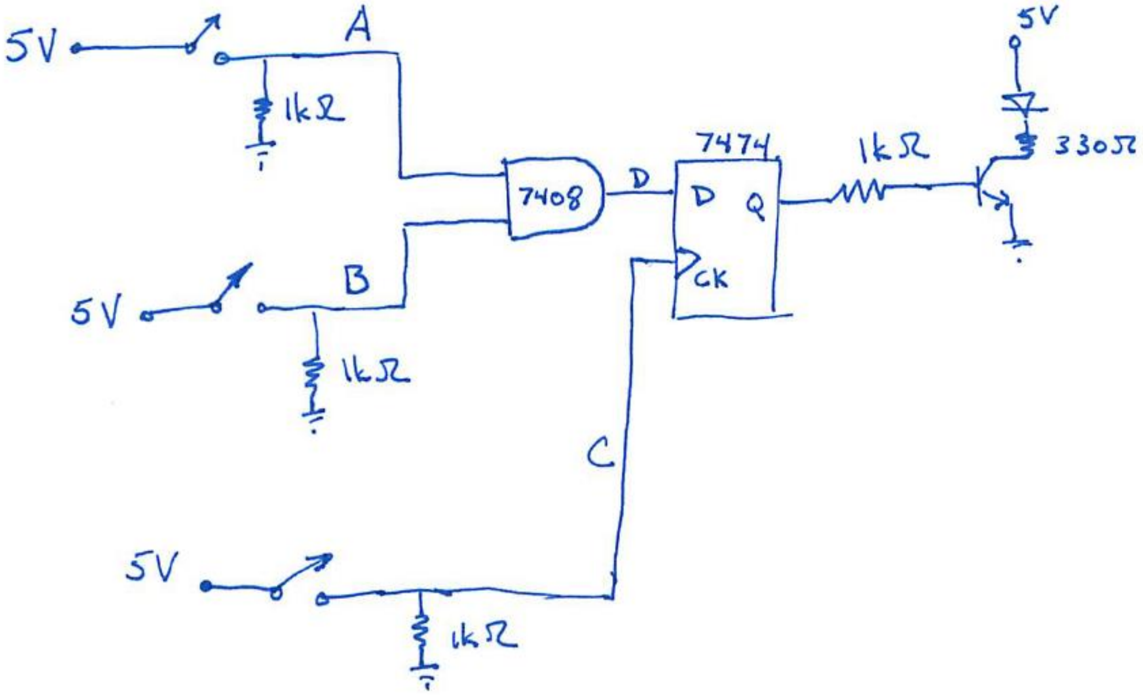


Figure 10.7 Circuit with switches, logic gate, and positive-edge triggered flip-flop.

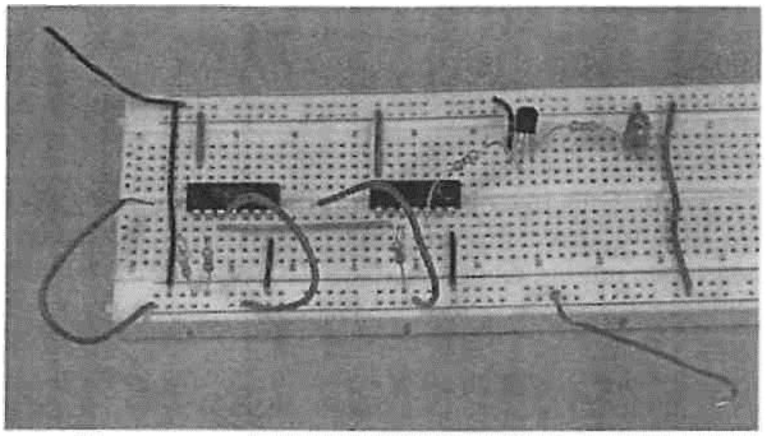
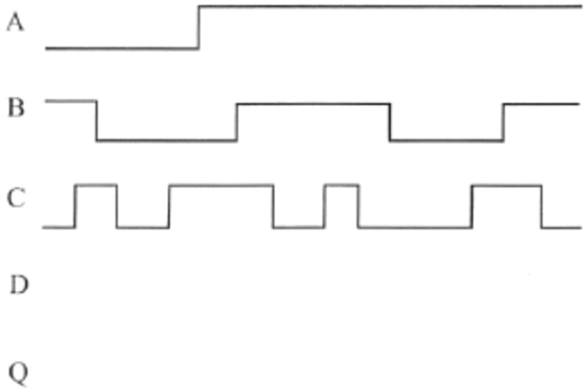


Figure 10.8 Photograph of the partially complete circuit.

Complete the following timing diagram and verify the results by testing your circuit. **Show me that your circuit is working properly before continuing.**



3. Replace the 7474 with the 7475 bistable data latch and rewire the circuit based on the pin-out diagram in Fig. 10.6. The circuit schematic is shown in Fig. 10.9. The only difference from the previous circuit is that the DataLatch is not edge triggered.

   NOTE - When removing ICs from a breadboard, always use a "chip puller" tool to lift both ends together. Alternatively, use a small flat-head screwdriver to pry each end up a little at a time to release the IC without causing damage (e.g., bent or broken pins).

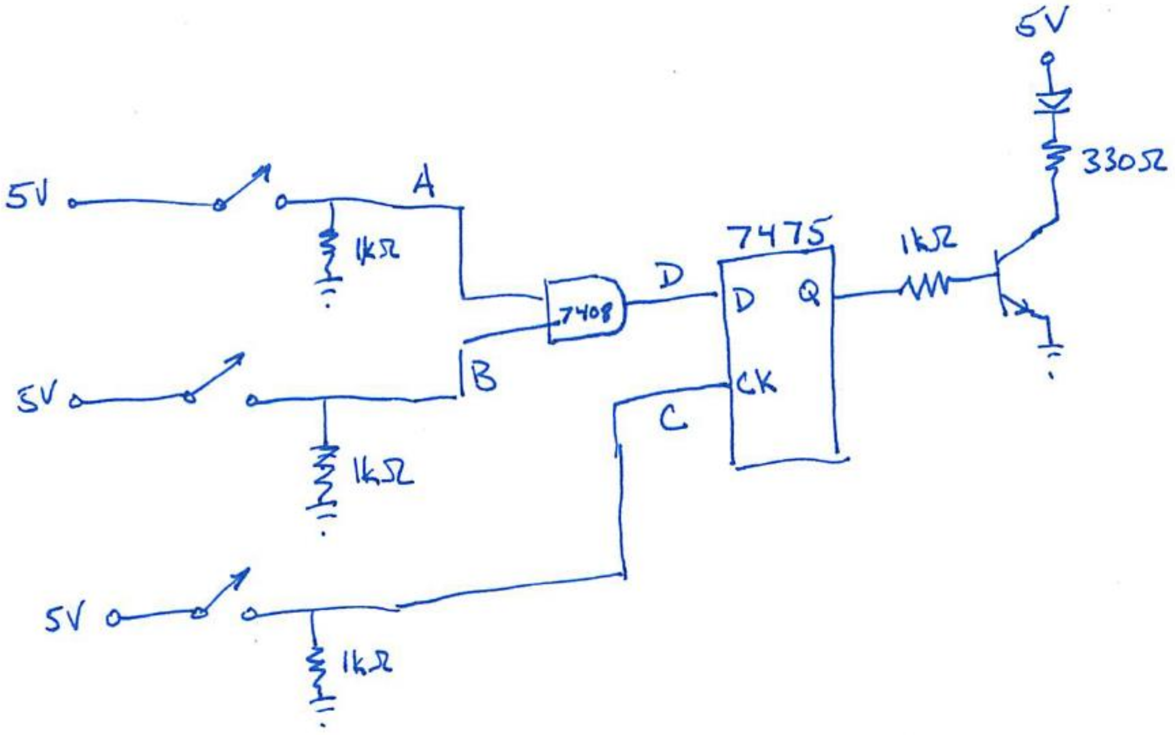


Figure 10.9 Data Latch Circuit (level triggered)

Complete the following timing diagram and verify the results by testing your circuit. **Show me that your circuit is working properly before continuing.**

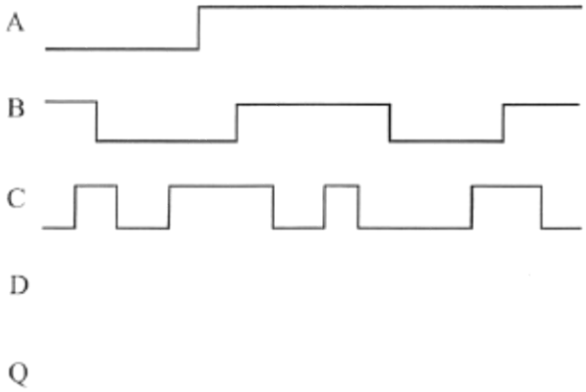


Figure 10.10 Latch circuit timing diagram.

LAB 10 QUESTIONS

Names:____________________________

1. Explain the difference between the output of the two circuits you analyzed and tested. What is the reason for the difference?

2. Switches and buttons often experience switch "bounce," especially when contact is made (as opposed to broken). Did bounce affect the output *Q* of the circuits? If so, in what cases?

   If there were bounce during the button *C* release, what would you expect? Draw modified timing diagram for each circuit (based on Fig. 10.10) to back up your claims.

   Would you expect to see any effects in the LED display, based on the timing diagram in Fig. 10.10, in either circuit if there were bounce during both the presses and releases of switches *A* and *B* (but not C)? Why or why not?

3. What is the purpose for the resistors between the switch outputs and ground?

4. Attach the detailed wiring diagram you used to construct the 7474 circuit. Make sure all of the pins used are labeled and numbered.