Lab 02 - Dealing with Imperfect Inputs

Introduction

Download the Word file with the assignment for this lab from the Lab Wiki page. Use it as you gather data in the experiments that follow.

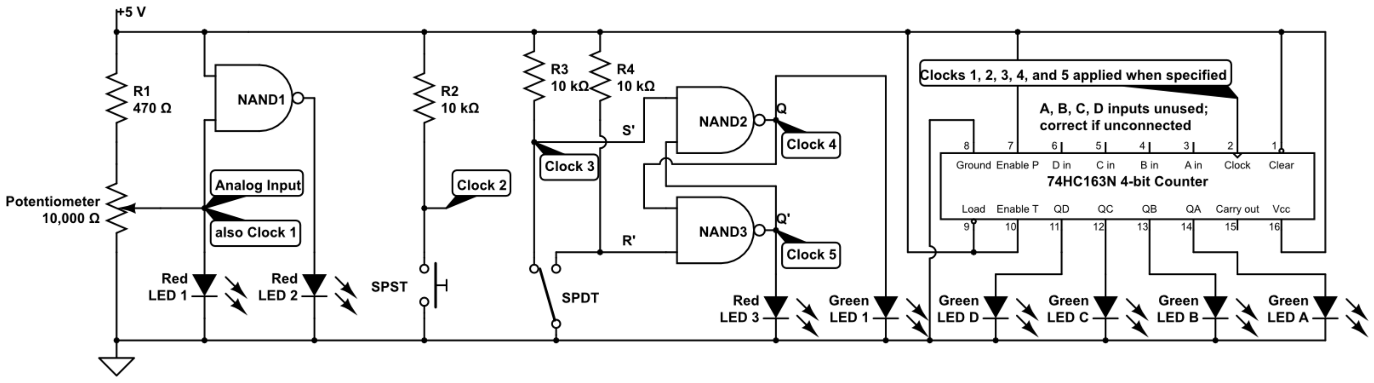
The real world is messy. Perfect digital signals do not exist. Sometimes a signal may be weak, and every signal has some amount of noise (random fluctuations) in it. Sources of noise in computer circuits are many, including fluctuations of power supply voltage, mechanical input switches, even cosmic ray bombardment causing ionization and extraneous electric charge a device in a circuit.

What to do? By examining the phenomena closely, techniques have been developed to restore weak signals before they degrade into uselessness and to overcome various types of noise. In this lab, we will observe how gates in our lab kit respond to input signals that are intentionally weak. We will use the 74163 4-bit synchronous binary counter to observe "bouncy" inputs from mechanical switches and then use an SR latch to debounce the switches.

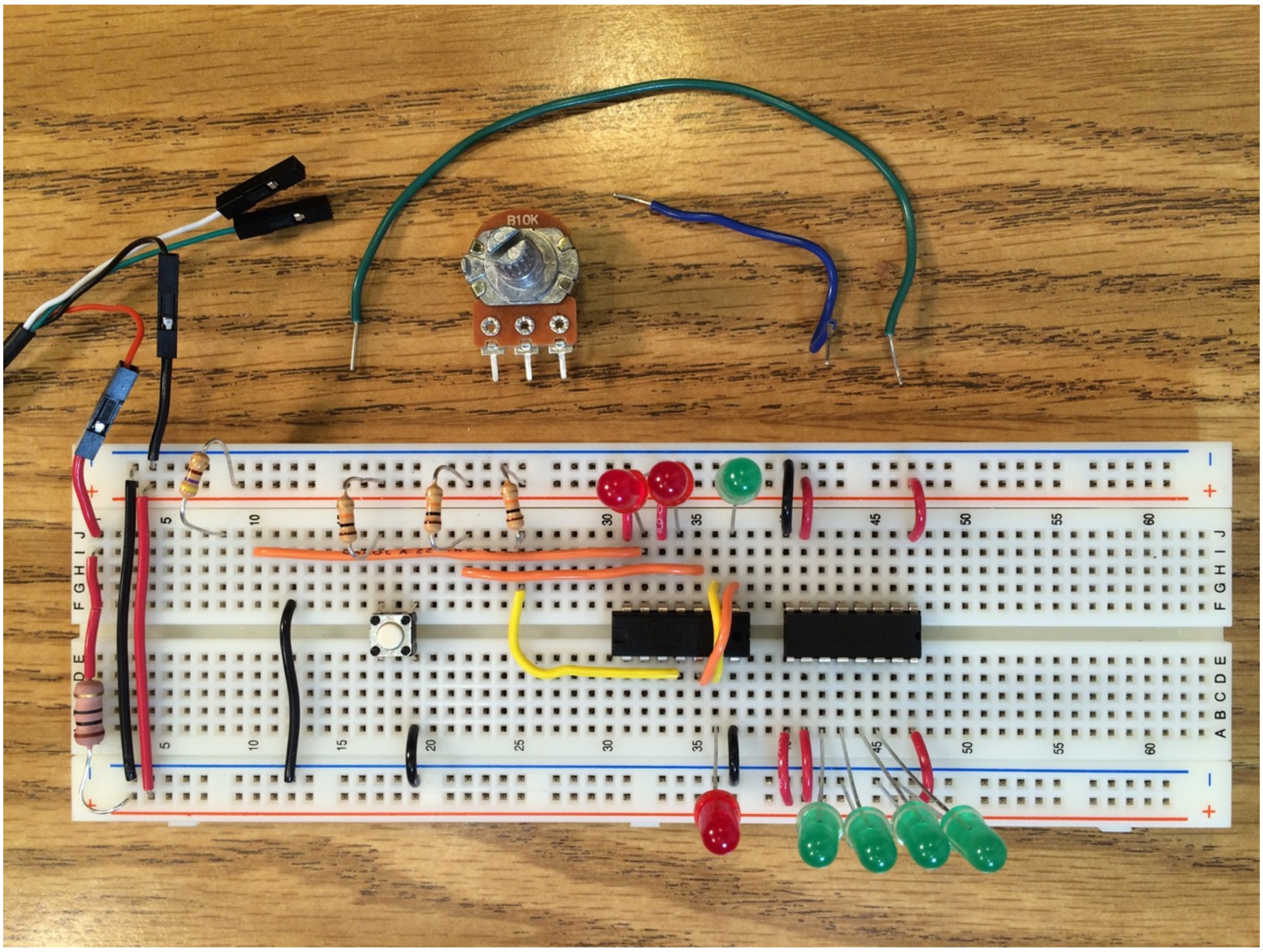
Grading for this lab is based on showing your TA your working circuit and answering the questions on the grading sheet.

Preparation

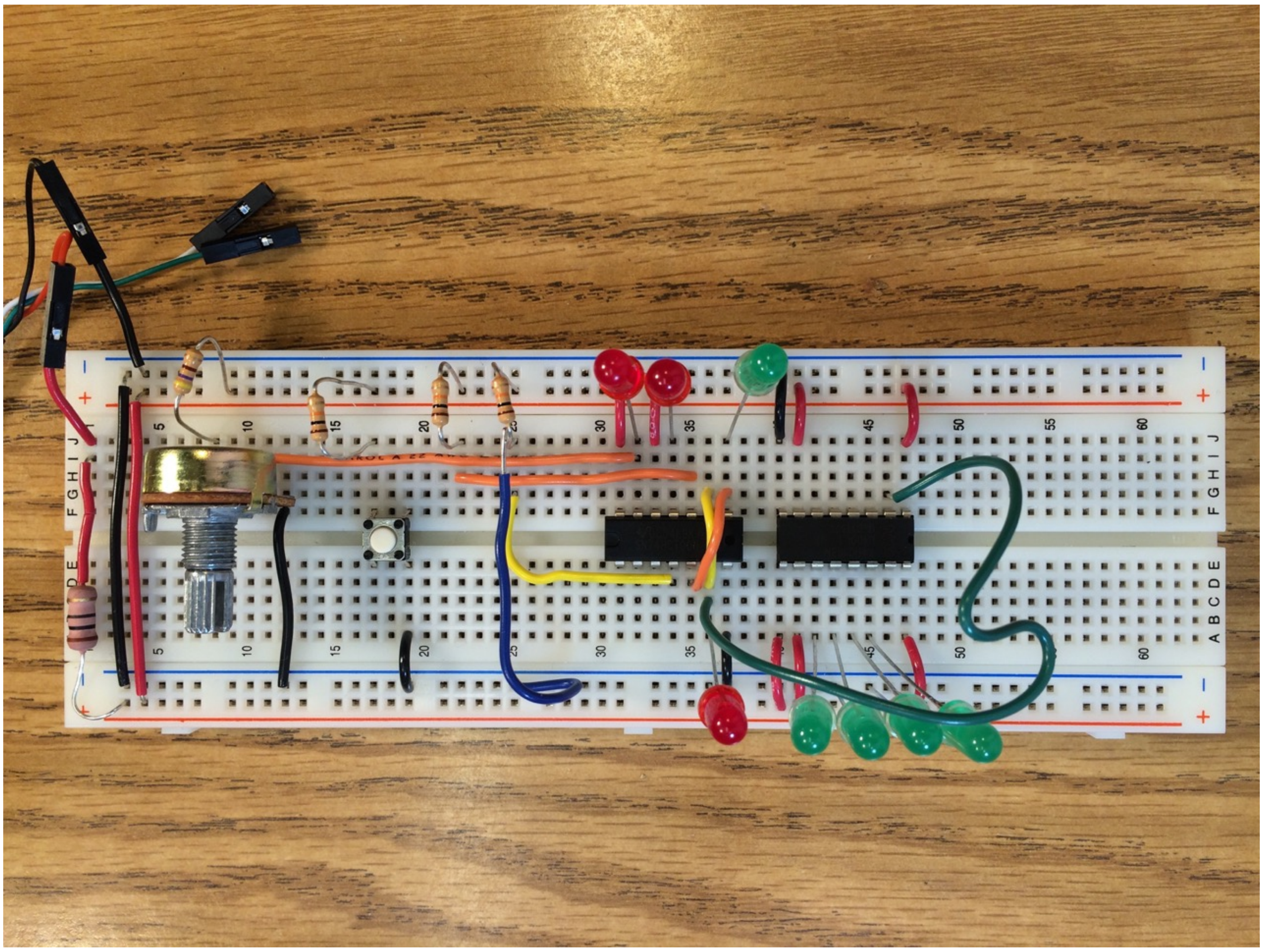
Build the following circuit on your breadboard.



Your result might look like this photograph before you add the potentiometer (which obscures some of the wiring as viewed from above) and the two final wires lying next to the breadboard in this photo. The 7400 NAND chip is placed with its label right side up in this photo. NAND1 of the schematic uses pins 13, 12, and 11 of the 7400; NANDs 2 and 3 are at the right end of the DIP. The 74163 is placed so that its label is upside down in the photo. This is so that output pins 11, 12, 13, and 14 and Green LEDs D, C, B, and A will appear in the left to right order of most significant bit to least significant bit.



This photo shows the breadboard with the potentiometer (its center pin in the column 10 sockets next to the orange wire), the blue moving wire that is the single pole of both the SPST and SPDT switches in column 25, and the green wire to carry clock signal to the 74163 from any of the 4 different input circuits.

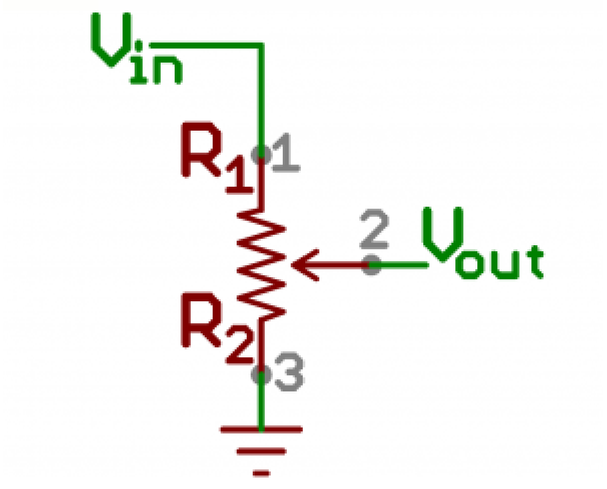


The 74163 is a presettable, synchronous, edge-triggered 4-bit binary counter with synchronous reset circuit. This mouthful of characteristics make the 74163 easy to use in a wide variety of circuit designs. We will use its edge-triggered design to count the number of rising edges, transitions from low to high, generated by 4 different input circuit designs. Typically, the 74163 can count rising edge events at 40 million per second. This is much more than fast enough to detect the behavior of the input switches we will build.

Experiments Part 1: Digital Gate Reaction to Analog Input

Voltage dividers in digital circuits are designed to provide two distinct outputs, high voltage or low voltage, falling within the acceptable bands for logic 1 or for logic 0. The 7400 family of circuits uses 5 volts to represent logic 1 and 0 volts to represent logic 0. But what is the real voltage on the wire? Can a logic 1 be 4.9 volts and still work properly? Can a logic 0 be 0.1 volts and still work properly? The answer is yes, and there is actually a whole range of input voltages which correspond to logic 1 and logic 0 (see powerpoint from lecture 2, slide 16 and the 7400 data sheet for details).

A potentiometer is a physically long resistor with the usual contacts at each end plus a moveable contact that slides along the length of the resistance element via a rotating knob or other mechanism. The moveable contact can be used as the output node of a voltage divider. Moving the contact changes the relative values of R1 and R2 shown in the schematic below, but the total R1+R2 remains the same. When the potentiometer is connected to Vin at terminal 1 and ground at terminal 3, the total voltage Vin must drop between terminals 1 and 3. This means that by moving terminal 2 from 1 to 3 the value of Vout can be varied from Vin (at position 1) to 0 (at position 3). The resistive element in the lab kit is linear. That means that as contact 2 moves from terminal 1 to terminal 3 the resistance between 1 and 2, R1, increases linearly with the increasing movement of terminal 2 towards terminal 3. So Vout will vary linearly with the position of terminal 2. What will happen when we send this linearly varying voltage to a 7400 NAND gate designed for a digital voltage input?



Experiment Data Gathering and Reporting

Download the Word document for this lab from Blackboard. Use it to record the data you gather in this portion of the lab.

In constructing the schematic circuit the 470 ohm resistor is there just to stop Red LED1 from being driven too hard by a full +5 V when terminal 2 of the potentiometer is at the extreme setting next to terminal 1.

[5 points] As you turn the potentiometer knob to move terminal 2 from terminal 1 (connected to the 470 ohm resistor) towards terminal 3 (connected to ground), the Analog Input voltage linearly changes from 5 volts to 0 volts. Carefully observe Red LED1 and Red LED2. Think about what Red LED1 and Red LED2 reveal about the operation of the potentiometer and NAND1. Use your observations to answer Question 1 in the lab assignment document that you downloaded already, and will upload into Blackboard for grading.

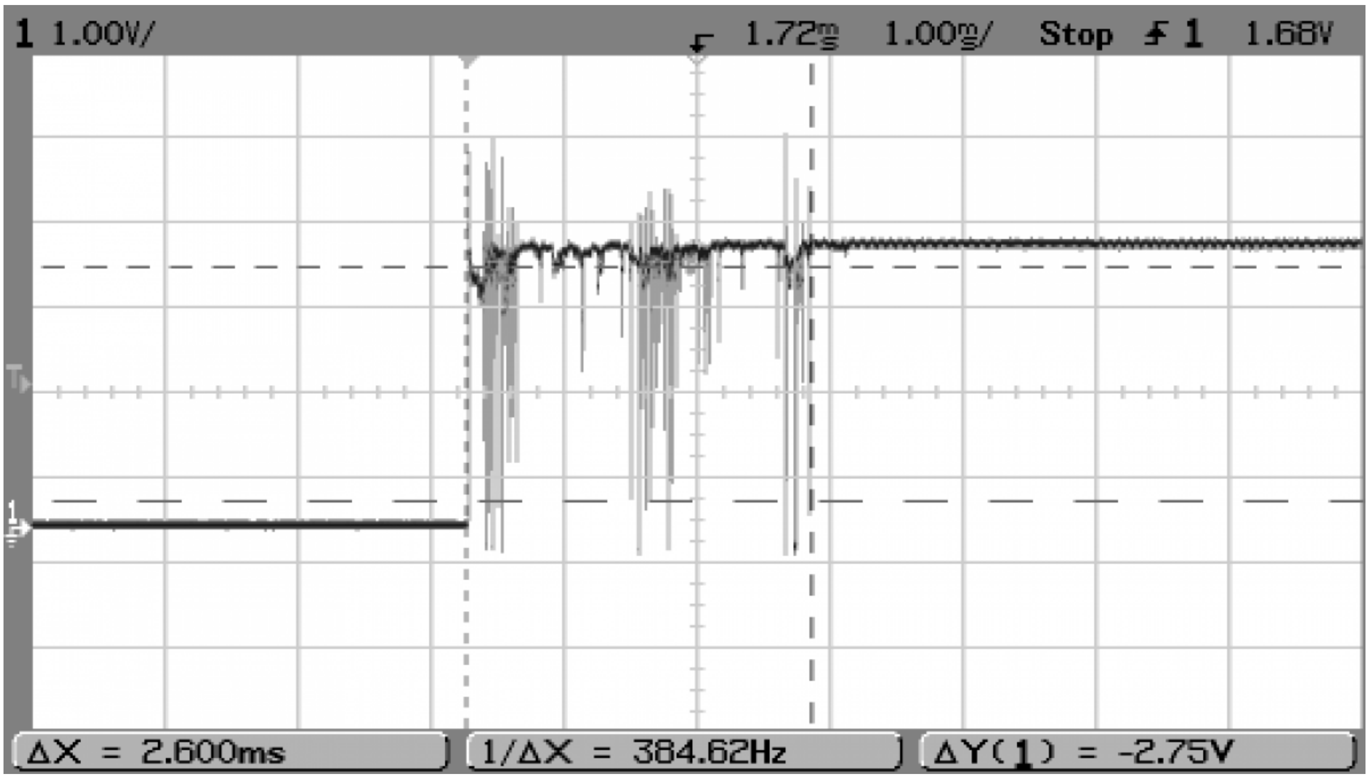
[15 points] Now use the lab voltmeters (your GTA will provide one) to measure voltage data pairs (NAND1 input voltage from potentiometer, NAND1 output voltage) at sufficiently many positions of the potentiometer to obtain a plot of NAND1 output voltage (y-axis) versus NAND1 input voltage from potentiometer. In particular, you should try not to miss gathering a data point just before LED2 switches from off to on, a data point when LED2 is partially on (may be tricky to get a potentiometer setting for this), ad a data point just after LED2 switches on. Ask the TAs in lab for directions on how to use the voltmeter if you have questions. Plot your data points in a graph to insert into the lab assignment Word document to answer Question 2.

Now connect the node named Clock 1 to the Clock input of the 74163 counter. Turn the potentiometer knob back and forth to generate rising edges for the 74163 to count. Look as Green LEDs D, C, B, and A to monitor the count as it changes. Can you find the potentiometer “sweet spot” where small back and forth motion is enough to change the count? Can you “trick” the 74163 into counting more than once for a small positive edge? Record your observations in the Lab Assignment document in the table for Question 4.

Experiments Part 2: Switch Bounce

The Lab Assignment document on Blackboard details the questions to answer from your work in Part 2 of this lab and the points for each question.

In switches with metal contacts the electrical resistance through the switch is often quite unstable for a few thousandths of a second as the contacts grind or bounce microscopically before settling to stable physical contact. This causes the output voltage of the switch to transition between voltage levels any number of times. Below is an image from an oscilloscope of signal voltage (vertical axis) versus time (horizontal axis) made as it measured voltage at a switch output as the switch closed. As you can see by the jagged line representing voltage, there are many transitions between the lower initial value and the higher final value before settling to the high voltage. If this signal was powering one of our LEDs, that LED would flash on and off in time with the signal because the response speed of an LED is quite high. Our own eyes however would not notice any of the flickering, just the final result achieved in a few thousandths of a second. If this signal was driving a digital logic gate, that gate would also respond to all of the unintended transitions. The problem is that it is not possible to build a mechanical switch that never exhibits these spurious voltage transitions. Our goal for Part 2 is to investigate bouncing in different switch designs. We will also investigate a hardware solution to the bouncing switch.



[Image credit: This image taken from post #12 of 22 at http://www.mytractorforum.com/24-gravely/259067-new-life-816-gravely-w-electronic-ignition.html by MTF Member # 51208 per fair use for educational purposes.]

We will use the 74HC163N 4-bit edge-triggered counter to count the number of positive voltage transitions made by a bouncing signal. Each rising edge of a signal delivered to the Clock input of the 74163 will increment the count by 1. The 74163 counts modulo 16, so when the count is 15 (when QD, QC, QB, and QA equal 1111), the next increment takes the count to 0 (QD, QC, QB, QA equal 0000).

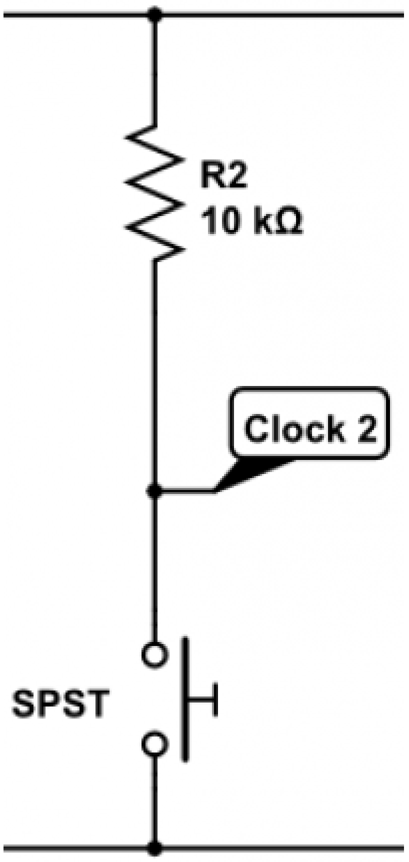
74HC163 is a 4-bit counter: http://www.futurlec.com/74HC/74HC163.shtml. A pin diagram is available through the link. How should the inputs be set so that the 74163 is in the mode to count, rather than one of its other modes: reset (clear), disabled (several types), and load 4-bit value into the counter?

Name	Purpose
POWER	As usual, Pin 16 and Pin 8 are corresponding to Voltage Supply(Vcc) and Ground(Zero reference, or GND).
CLOCK INPUT	Pin 2 is the clock input pin.
DATA INPUT	Pin 3 to Pin 6 are data input pins corresponding to D0 to D3.
COUNT ENABLE	Connect Pin 7 to Vcc to enable the counting function.
FLIP-FLOP OUTPUT	Pin 14 to PIN 11 are the output pins. They are able to maintain the output by using flip-flop mechanism.
COUNT ENABLE CARRY INPUT	Connect Pin 10 to Vcc if we want enable carry input function.
PARALLEL ENABLE INPUT	Connect Pin 9 to Vcc to disable the function.

2a: Single Pole Single Throw (SPST) Switch made with a Push Button

Connect Clock 2 to the Clock input (pin 2) of the 74163 counter. Press the button, counting from 0 to 15, and note the frequency and amount of skipped numbers on the 4-bit binary counter. The push button switches are not very bouncy, especially when generating low-active inputs (Why?). You may need to go through a number of cycles of modulo 16 counting before you see a count skip indicating bouncing occurred. Enter your data in the lab Grading Form (link at the bottom of this page).

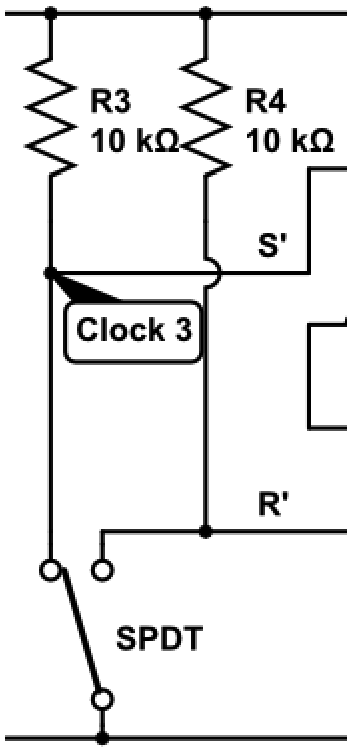
In the picture below, a push button is used for the switch.



2b: Single-Pole Single-Throw (SPST) Switch made with Wire

Next, build a SPST switch using only one side of the already constructed single-pole double-throw (SPDT) switch (shown in the portion of the breadboard schematic below). Simply connect the counter input to Clock 3 (the left side of the SPDT) which is the orange wire in row 22 in the photo. Touching the blue wire to the bottom of the 10 Kohm resistor is the same as pressing a button switch, and not touching the wire to the resistor is the same as not pressing the button, so SPST. Touch the two wires together 16 times, counting from 0 to 15, and note the frequency and amount of skipped numbers on the 4-bit binary counter. You should see lots of skipping; this is a very bouncy switch design. Enter your data in the lab Grading Form (link at the bottom of this page).

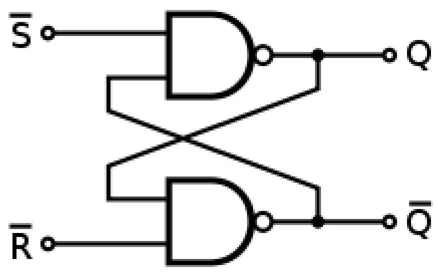
In the schematic below, a wire is used for the switch.



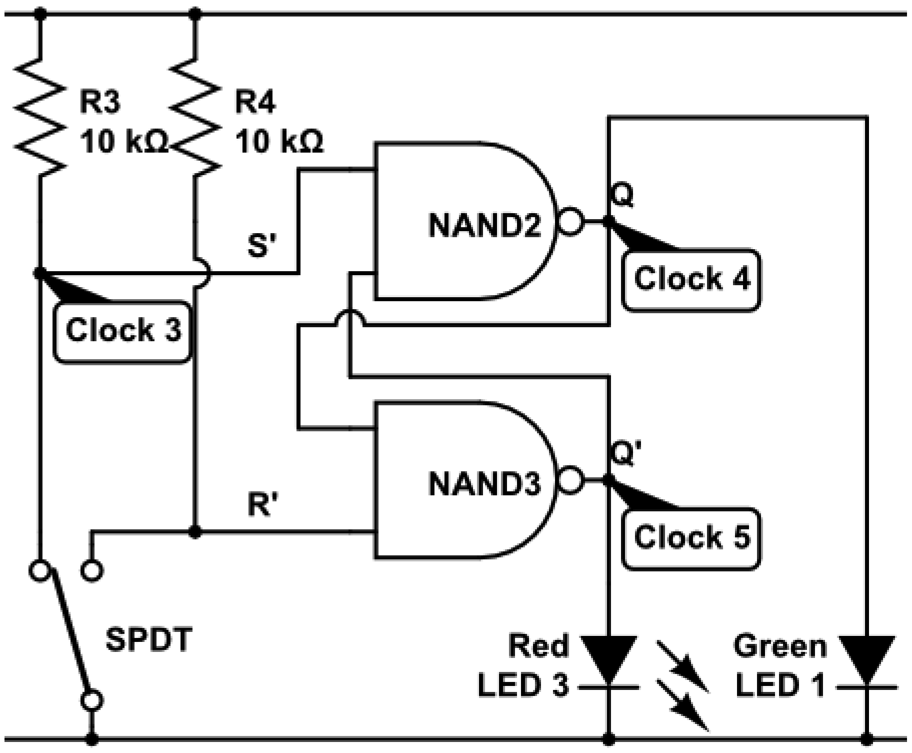
2c: Single Pole Double Throw (SPDT) Switch

Next, use the SPDT switch to generate input to the counter (Clocks 4 and 5). We know that the SPDT switch bounces by virtue of testing part of it as the SPST switch. Now, using NANDs 2 and 3 we built an SR latch to prevent the switch bouncing from propagating to the 74163 clock input. The truth table for an SR latch is shown below. Truth tables for an SR latch usually use inputs labeled **S'** and **R'** for “set” and “reset”. The inputs are labeled inverted because they are active low inputs. The outputs are labeled **Q** and **Q'** because they opposite in normal operations (Normal being all the time we do not ask the latch to simultaneously Set and Reset; be nice.). An SR latch is the basic building block for computer memory (very high speed, but also expensive and energy hungry). The latch will hold the current Q value as long as both inputs are logic 1. In the schematic of the SPDT switch, the two 10 Kohm resistors hold the inputs to NANDs 2 and 3 to logic 1. To “set” or “reset” the output of the SR latch, the corresponding input must be set to logic 0 for a long enough for the NAND gate to notice and react, that is, for around 10 nanoseconds. For example, if you want to set the output of the SR latch to logic 1, ground the **S'** input by touching the blue wire in the SPDT switch to the left resistor. The **Q** output (Clock 4) will be logic 1, even if you bring the **S'** input back to logic 1 by disconnecting the wire from the resistor. Can you see how the bouncing of the metal to metal contact is masked by the SR latch?

Input		**Output**	
S'	R'	Q	Q'
0	0	Not valid	
0	1	1	0
1	0	0	1
1	1	Keep old value	



Touching the blue wire to the right resistor will send 0 volts to the input of the bottom NAND gate, causing the Red LED3 to turn on. Touching the blue wire to the left resistor will send 0 volts to the input of the top NAND gate, causing Green LED1 to turn on.



Now connect Clock 4 to the Clock input of the 74163 and gather data on the performance of this SPDT and latch combination by touching the blue wire back and forth across the two resistors, setting and resetting the output 16 times, and note the frequency and amount of skipped numbers on the 4-bit binary counter. Enter your data in the Grading Form. What changes if you connect Clock 5 to the 74163?

Part 3: Take-home and Grading

See Blackboard to download the Word file for the take home portion of this lab and the grading information.

You will design a simple logic circuit in SOP and POS form to complete this lab.