**Lab 02 - Coping with Imperfect Inputs**

**Upload your answers to the following 11 Questions in PDF to Blackboard before the start of your lab session next week.**

**Answer these six questions during your lab session this week (before you leave lab).**

1. [5 points] As you turn the potentiometer knob to move terminal 2 from adjacent to terminal 1 (connected to the 470 ohm resistor) towards terminal 3 (connected to ground), the voltage at the circuit node labeled Analog Input linearly changes from 5 volts to 0 volts. Carefully observe and then describe here, the illumination behavior of Red LED1 and Red LED2, as you turn the knob.

As we turn the potentiometer knob from terminal 2 to terminal 1. At terminal 1 the led 2 lights up and at terminal 2 led 1 lights up.

1. [5 points] What do Red LED1 and Red LED2 reveal about the operation of the potentiometer and about the operation of NAND1?

The potentio meter sends an analogue input to the led 1 but so that is why it gradually dims and then turns off. But the led 2 gets output from nand1 that is a digital circuit and that is why it fully lights up.

1. [15 points] Use a lab multimeter (your GTA will provide one) to measure voltage to ground, in data pairs, for the following pair of points (nodes) in your circuit: (Analog Input, NAND1 output pin).

|  |  |
| --- | --- |
| Analog | Nand Out |
| 2.7 | 0 |
| 2.29 | 0 |
| 1.80 | 0 |
| 1.69 | 0 |
| 1.65 | 0 |
| 1.60 | 0 |
| 1.50 | 0 |
| 1.44 | 0.90 |
| 1.30 | 4.20 |
| 1.10 | 4.19 |
| 0.40 | 4.18 |
| 0 | 4.19 |

**It can be difficult to gather voltage measurements, record them, and rotate the potentiometer knob all with just two hands. Therefore, for this lab it is acceptable that you work with another CS250 student to make these voltage pair measurements. One person might hold the probes of the multimeter while the other person incrementally rotates the knob and records the result. You may share voltage data pairs with another student.**Obtain voltage pairs at sufficiently many rotational positions of the potentiometer knob to yield a nice smooth plot of NAND1 output voltage (y-axis) versus NAND1 input voltage from potentiometer (x-axis). In particular, gather enough a data points around the time LED2 switches from off to on.  
  
Plot your data point pairs (Analog Input, NAND1 output voltage). Comment on the shape of the function NAND1 output voltage = f(Analog Input) displayed in your plot. Does f() show digital behavior?

Fill in this table with your observations from Questions 4, 5, 6, and 7. Clearly describe what you observe.

|  |  |
| --- | --- |
| **Clock Source** | **Number of bounces observed on LEDs D, C, B, A** |
| Clock 1 (rotating potentiometer) | Sweet spot when increments by power of 2 |
| Clock 2 (SPST pushbutton switch) | (5 and 14) incremented twice |
| Clock 3 (half of SPDT bare wire switch) | 3,5,3,1,5,3,9,8,5,4,4,6,11,3,4 |
| Clock 4 or 5 (Q or Q’ output of SR latch) | 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 |

1. [5 points] Connect the circuit node named Clock 1 to the Clock input of the 74163 counter using the green wire. Turn the potentiometer knob back and forth to generate rising edges for the 74163 to count.
   1. Which direction, clockwise or counterclockwise, generates a rising edge?

clockwise

* 1. Observe 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 rotational motion is enough to change the count? Record your observations in the Clock Source table.

1. [10 points] Press the pushbutton switch 16 times to generate Clock 2. Record the amount that the count increases each time in the Clock Source table. The push button switches have a design that bounces rarely. Keep pressing the pushbutton, while counting, until you see the count increase by more than one. Record the number of pushes you made before you saw a bounce
2. [10 points] Touching the blue wire to the bottom of the 10 Kohm resistor is the same as pressing an NO SPST button switch. Not touching the wire to the resistor is the same as not pressing the button. Touch the two wires together 16 times to generate Clock 3. Each time you touch, record in the Clock Source table the increment amount modulo 16 you see on the 74163 output (LEDs D, C, B, A).
3. [10 points] Now connect Clock 4 to the Clock input of the 74163 and gather data on the performance of the bouncy SPDT switch and SR latch combination by moving the blue wire back and forth between R3 and R4, setting and resetting the latch output 16 times. Record in the Clock Source table the increment amount modulo 16 that you see on the 4-bit binary counter each time.
   1. What changes if you connect Clock 5 to the 74163 and again operated the SPDT switch as before?

The increment is the same as before.

* 1. Is the 74163 count behavior the same whether you input 16 pairs of S’R’ or 16 pairs of R’S’?  
     Yes

**Answer the following four questions before your lab session meets next week.**

**Upload your answers to Questions 1 through 11 to Blackboard before your lab session meets next week.**

1. [10 points] Why is skipping consecutive numbers in the output of the 74163 counter an indication of switch bounce?

When we press the switch, there comes a time when the switch bounces back and forth against the wires, this connection and disconnection (bounces) is too fast for our eyes to notice but the 74163 counter counts these bounces and they appear to us in the form of skipped numbers.

1. [10 points] If the 74163 advances the count by 1 for a single Clock 3 input this means that the switch did not bounce that time. True or False? Explain your answer.

It is false because even though the counter only incremented by one every time it could have incremented by a multiple of 16 and our eyes would still be too slow to catch it. Hence, we cannot say that it bounced at that time.

1. [10 points] What is the mathematical expression for the number of bounces observed by the 74163 chip circuitry as contrasted with the number of bounces that your eyes are capable or observing by examining the 74163 output using LEDs?

Observed number of bounces [mod 16]

1. [10 points] How does the memory capability of the SR latch (NAND2 and NAND3) transform the bouncing SPDT switch input into a bounce-free latch output Q?

This can be achieved by using the set reset and storage capabilities of the SR latch. In our resistors, the S and R inputs are at a logic one when no contact is made by the switch. When the switch is normally connected to the S’ input to the latch, making S’ as 0 and Q as a logic 1. We are setting the latch to one. Pushing the switch changes these inputs from S’ as 0 and R’ as 1 to S’ as 1 and R’ as 1.

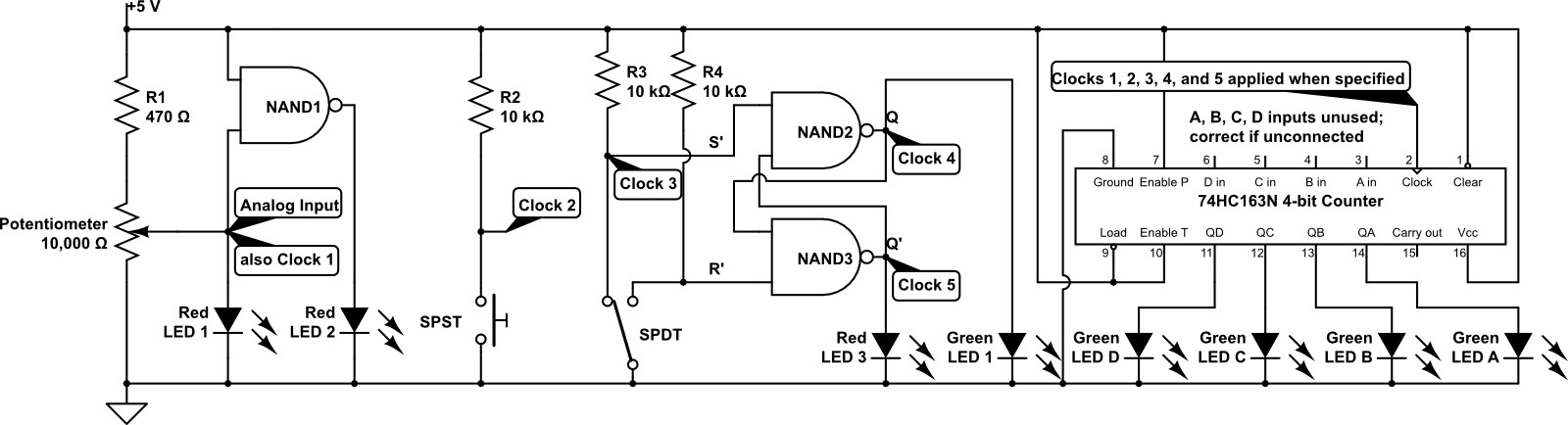
# Lab Introduction

The real world is messy. In actual hardware logic signal voltages differ somewhat from ideal. In extreme cases, a voltage may fall outside the intended acceptable bands for the two logic levels. Sometimes the voltage of a signal may not hold steady at the intended value.

What happens with out-of-band voltages and what to do about some unsteady signals? In Part 1 this lab, we will observe how a NAND gate in our lab kit responds to an input signal with voltages that are intentionally outside the acceptable bands for logic 0 and logic 1. In Part 2 we will use the 74163 4-bit synchronous binary counter to observe signals where the voltage does not hold steady for a few thousandths of a second because of “bouncy” mechanical switches in the voltage divider. Then we will see how an SR latch can de-bounce the voltage divider signal even though the mechanical switch still bounces.

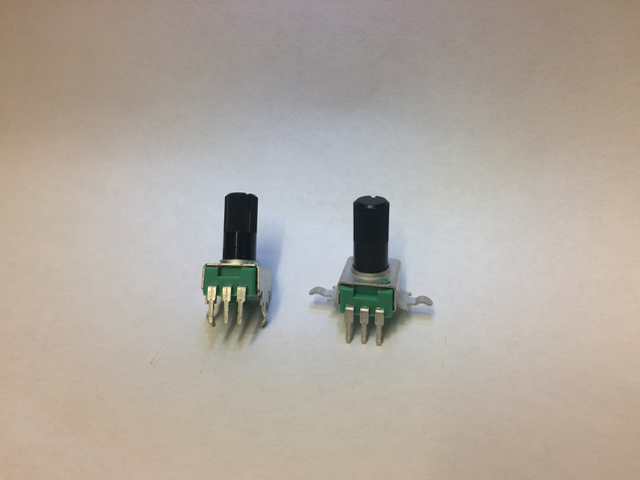
# Preparation

Build this circuit on your breadboard.



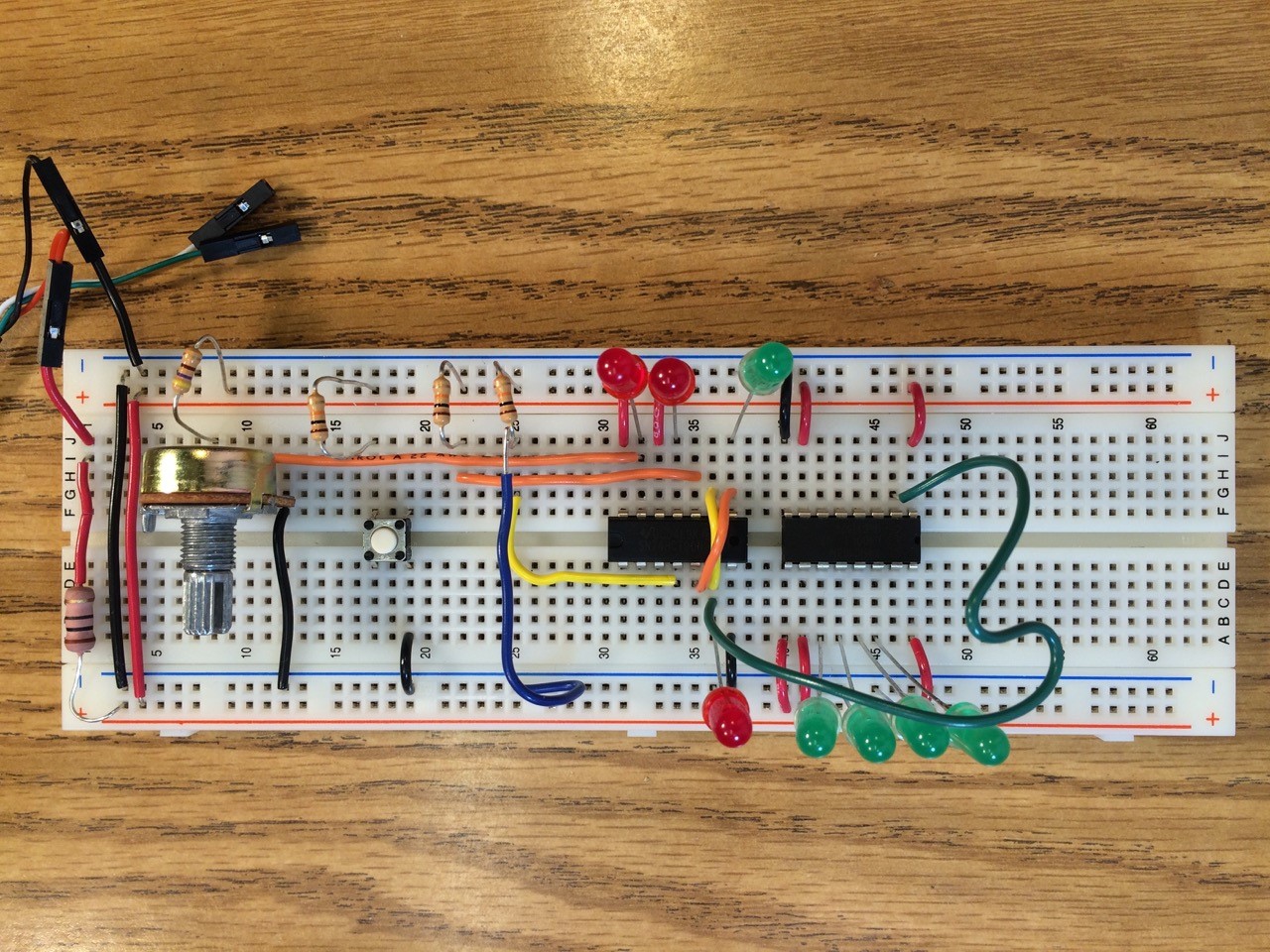
Your result might look a bit like the following photo. The 7400 NAND chip is placed with its label right side up and its notch facing left in this photo. NAND1 of the schematic uses pins 13, 12, and 11 of the 7400; NANDs 2 and 3 are at the right end of the DIP. The 74163 is placed so that its label is upside down and its notch facing right. This is so that output pins 11, 12, 13, and 14 and Green LEDs D, C, B, and A will conveniently appear in the left to right order of most significant bit to least significant bit.

The potentiometer in the breadboard photo is a different design than was purchased for the lab kits this year. Here is a photo of this year’s green-bodied and black-knobbed potentiometer. Use your needlenose pliers to bend the two big tabs of your potentiometer until they are at right angles to the axis of the black knob.



Then insert your potentiometer into your breadboard with the black knob pointing vertically upward from the board, the body of the potentiometer over the gutter of your breadboard, and with potentiometer terminals 1, 2, and 3 in breadboard points F08, F09, and F10. If this space is not open on your breadboard then place the potentiometer in a nearby similar position. This positioning should make it easy to connect R1 and wires to the potentiometer terminals. Terminal 2 of a potentiometer is always the middle one of the three pins.

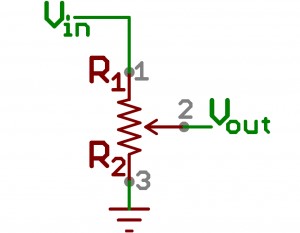
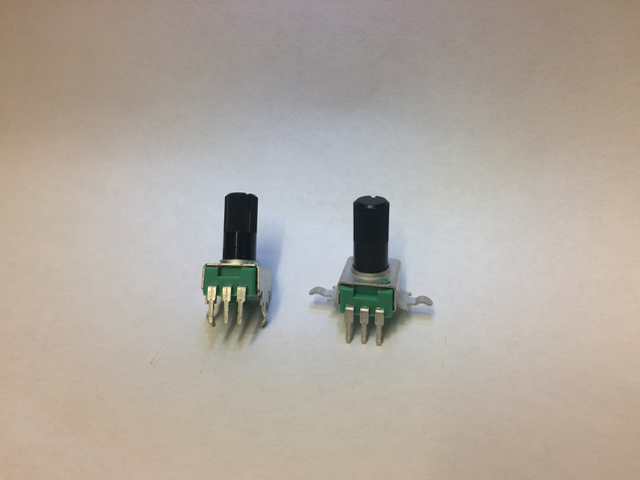
The blue wire in breadboard ground in row 25 is intended to touch the leads of R3 and R4 one at a time as it is moved back and forth. The long green wire must be able to connect pin 2 of the 74163 chip to the five circuit nodes labeled Clock 1 through Clock 5 in the schematic.



# Experiment Part 1: NAND gate reaction to an analog voltage 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. Can a logic 1 be 4.9 volts and the NAND still work properly? Can a logic 0 be 0.1 volts and operation be proper? The answer is yes; there is range of input voltages which correspond to logic 1 and logic 0 (see lecture 2, slide 16 and the 7400 data sheet for details).

A potentiometer is a resistor with the usual leads at each end (1 and 3 in the schematic below) plus a moveable contact (2 in the schematic) that slides along the length of the resistance element. The moveable contact is connected to the middle lead on the device. 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 fixed. When the potentiometer is connected to Vin at terminal 1 and ground at terminal 3, the voltage Vin will drop to 0 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).



For the lab kit potentiometer, the resistor is laid out along a 270 degree circular arc and the resistance between terminals 1 and 3 is distributed linearly along the length of the arc. Thus, as the potentiometer knob is rotated, and contact 2 slides along from being adjacent to terminal 1 to being adjacent to terminal 3, the resistance between terminals 1 and 2, the R1 resistance in the schematic above increases linearly with the movement of terminal 2 towards terminal 3. Therefore, Vout at terminal 2 will vary linearly with the position of the contact connected to terminal 2 (always the middle pin of the three pins).

## Analog Input and Clock 1 data measurements

What happens when we send Vout that varies smoothly from 0 volts to +5 volts to a NAND gate input?

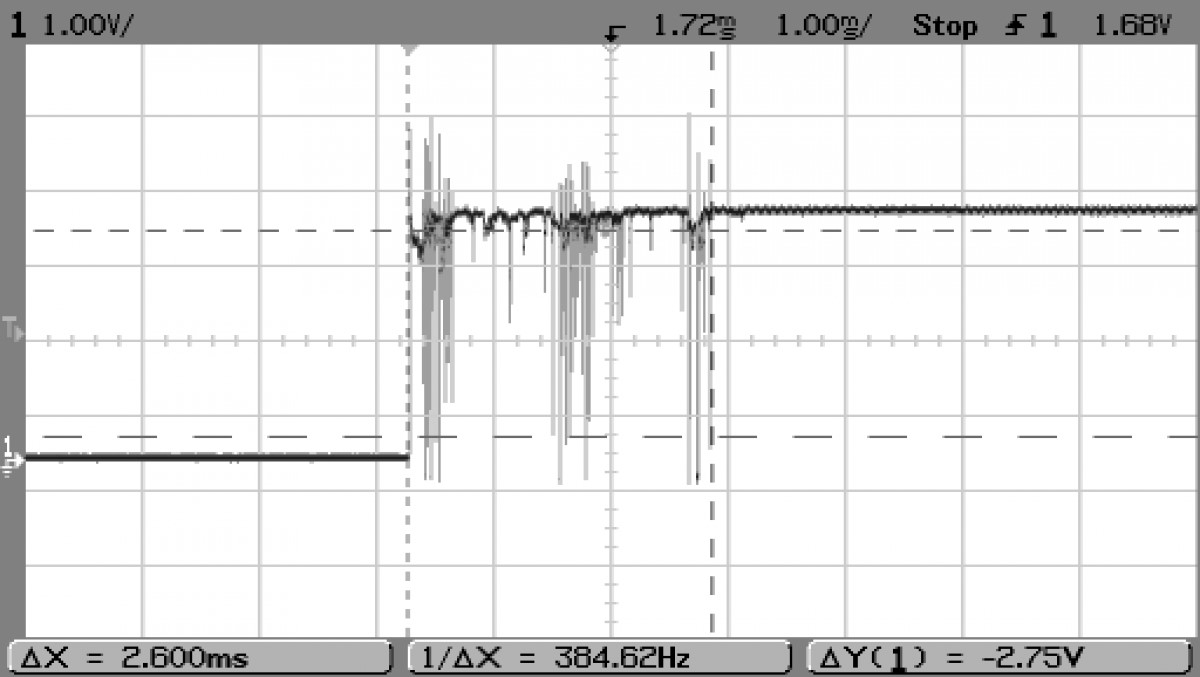
Answer Questions 1, 2, and 3 at this time. Note that R1 is in series with the potentiometer to prevent Red LED1 from being driven too hard by being directly connected to a full +5 V when terminal 2 of the potentiometer is at the extreme setting adjacent to terminal 1.

Connect Clock 1 to the clock input of the 74163 chip. Answer Question 4.

# Experiment Part 2: Mechanical switch bounce

The electrical resistance between the contacts of a mechanical switch is often quite unstable for a few thousandths of a second as the contacts grind or bounce microscopically before settling to a stable physical contact. This causes the voltage delivered to the switch output to transition between switch-open and switch-closed levels any number of times.

An oscilloscope is an instrument that can plot voltage as a function of time. Below is an image from an oscilloscope plotting switch output voltage (vertical axis) versus time (horizontal axis) as the switch is closed. The horizontal axis shows time over an interval of 10 milliseconds. The switch bounces for 2.6 milliseconds at the center of the time interval, before completely making or breaking contact, we cannot tell which is the case because switches can bounce at both times. The jagged transitions of voltage tend to be random in number and duration each time the switch makes or breaks contact.

[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.]

If this switch output was powering an LED, that LED would flash on and off in time with the voltage variations because the response speed of an LED to the voltage presented to it is quite fast. Human eyes, however, would not notice the flickering because it has too high a frequency, just as some sounds are ultrasonic in frequency and cannot be heard by humans but can be heard by bats. All we would see would be the final LED state after the switch stops bouncing between closed and open. Crucially though, if this signal is input to a digital logic gate that computes its output in mere nanoseconds, that gate would re-compute its output logic level with every input signal transition. This is unlikely to be the circuit behavior that we, as designers, want.

While clever mechanical design can reduce mechanical bounce, it is not possible to build a mechanical switch that never exhibits bounce. Our goal for Part 2 of this lab is to investigate bouncing in different switch designs. We will also investigate an electronic solution to the bouncing mechanical switch.

The 74163 chip <http://www.futurlec.com/74HC/74HC163.shtml> is a presettable, synchronous, rising-edge-triggered 4-bit binary counter with synchronous reset circuit. These characteristics mean that the 74163 will output a count, modulo 16, of the number of rising edges (transitions from low voltage to high voltage) appearing in the signal sent to its Clock input. We will move the green wire to various signal nodes on the breadboard, sampling clock signals from four different input circuit designs for rising edges.

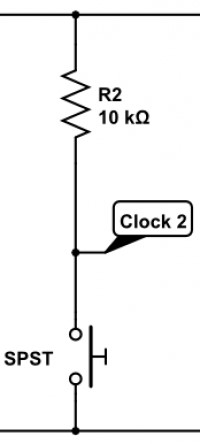
The 74163 can count rising edge events at a rate of up to 40 million per second, more than fast enough to “see” mechanical contact bounce. Each rising edge of a signal delivered to the Clock input of the 74163 will increment the count by 1. The 74163 counts modulo 16, so when the count is 15 (when QD, QC, QB, and QA equal 1111), the next increment takes the count to 0 (QD, QC, QB, QA equal 0000).

A pin diagram for the 74HC163 is available at <http://www.futurlec.com/74HC/74HC163.shtml>. How should the chip 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 using lab kit pushbutton switch: Clock 2

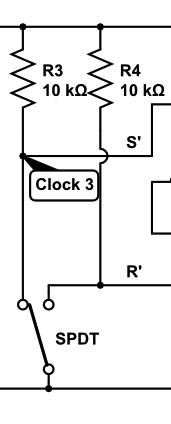
In the picture below, a push button is used for the switch.



Connect Clock 2 to the Clock input (pin 2) of the 74163 counter. Edit this document to add your answer to Question 5.

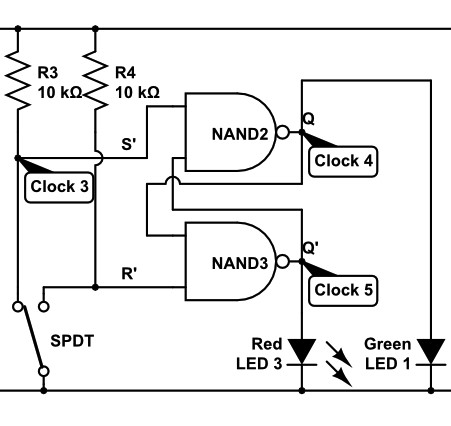
## 2b: Single-pole single-throw (SPST) switch made with bare wire: Clock 3

In the schematic below, a wire is used for the moving pole of the switch that can touch either of the two throws that are connected to R3 and R4. As implemented the bare wire leads of R3 and R4 serve directly as the two switch throws and moving the bare end of the wire to ground back and forth between these two lead comprises the moving pole. A crucial part of the mechanical design of this switch is that the pole (moving wire) cannot physically touch both throw wires at the same time. This switch always passes through a not-connected state when moving from touching one throw wire to touching the other throw wire.



Build an SPST switch simply by restricting your use of the already-constructed single-pole double-throw (SPDT) switch (shown in the portion of the breadboard schematic above) to just one of the two throws. 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. Now answer Question 6. You should see a wide variety of incremental count amounts by the 74163: touching a bare wire with another bare wire is a very bouncy mechanical switch design.

## 2c: Single-pole double-throw (SPDT) switch driving an SR latch for electronic de-bouncing: Clocks 4 and 5



NAND2 and NAND3 are connected to form an SR latch. The schematic also shows that R3 and R4 hold each of the S’ and R’ inputs to NAND 2 and NAND 3 to a logic 1 value, except when the SPDT switch pole makes contact with one of the two throws.

We know that each throw of the SPDT switch bounces by virtue of testing one throw as the SPST switch in part 2b. We can also see that by its mechanical design, which separates the two throws in space by more than the diameter of the moving pole wire, that the single pole cannot contact both throws at the same time. The mechanical design of this switch ensures that, in moving the pole from one throw to the other, there will be a time during which the pole makes no contact with either pole.

Here is the truth table for an SR latch. This latch design sustains (holds) the current Q output value as long as both inputs S’ and R’ remain at logic 1. However, when S’ becomes 0 the latch will set the Q output to 1 within a few nanoseconds and will reset Q to 0 nanoseconds after R’ becomes 0.

|  |  |  |  |
| --- | --- | --- | --- |
| S’(t) | R’(t) | Q(t) | **Q(t+Δ); comment** |
| 0 | 0 | 0 | X; Disallowed S’R’ input combination |
| 0 | 0 | 1 | X; Disallowed S’R’ input combination |
| 0 | 1 | 0 | 1; Set; make output Q equal to 1 |
| 0 | 1 | 1 | 1; Set; make output Q equal to 1 |
| 1 | 0 | 0 | 0; Reset; make output Q equal to 0 |
| 1 | 0 | 1 | 0; Reset; make output Q equal to 0 |
| 1 | 1 | 0 | Q(t); Hold or sustain the current output |
| 1 | 1 | 1 | Q(t); Hold or sustain the current output |

Mechanical switch bounce happens on a time scale of milliseconds, while NAND gates react to changes on their inputs on a time scale of tens of nanoseconds, which is about 100,000 times faster. Imagine that we touch the switch pole wire to the S’ input, grounding it (logic 0), and then one-thousandth of a second afterwards the wire bounces mechanically to not touching the S’ input, changing the S’ logic value back to 1. Long before the bounce returns S’ to 1, NAND2 will have responded to S’=0 with Q(t+Δ)=1. Then, in nanoseconds, the feedback path of Q to the input of NAND3 along with the fact that R’=1 will have caused NAND3 output Q’(t+Δ)=0, which is fed back to NAND2. There, at the inputs to NAND2, regardless of whether S’=0 or S’=1, the latch is stable with Q(t+Δ)=1. The NAND2-NAND3 latch circuit “catches” the first touch of the pole wire that makes S’=0 and ignores all pole wire bouncing on the S’ contact afterwards.

Touching the pole wire to R4 makes R’=0, changing Q to 0 and Q’ to 1 causing Red LED3 to turn on. Now answer Question 7. Before the start of your next week lab session, answer Questions 8 through 11.

**Upload your answers for all questions in PDF format to Blackboard before the start of your lab session next week.**